KNOWLEDGE-BASED INTERPRETATION OF REMOTE SENSING DATA WITH THE INTERIMAGE SYSTEM: MAJOR CHARACTERISTICS AND RECENT DEVELOPMENTS

G. A. O. P. Costa [a, *], R. Q. Feitosa [a], L. M. G. Fonseca [b], D. A. B. Oliveira [a], R. S. Ferreira [a], E. F. Castejon [b]

[a] Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro.
Rua Marquês de São Vicente, 225, Gávea - Rio de Janeiro, RJ – Brazil
(gilson, raul, dario)@ele.puc-rio.br, rodrigosilferreira@gmail.com
[b] Image Processing Division (DPI), National Institute for Spacial Research (INPE)
Av. dos Astronautas, 1.758, Jd. Granja - São José dos Campos - SP – Brazil
(leila, castejon)@dpi.inpe.br

ABSTRACT:

The InterIMAGE Project aims at developing an open-source, knowledge-based framework for automatic interpretation of remote sensing data. The Project, which began in 2008, is leaded by the Computer Vision Lab of the Catholic University of Rio de Janeiro in collaboration with the Brazilian National Institute for Space Research (INPE) and the TNT Institute of the Hannover Leibniz University. The interpretation strategy implemented by InterIMAGE is based on a user defined knowledge model, structured through a semantic network. The nodes of the network represent concepts, classes of image objects expected to be found in a scene. Specialized image processing operators can be attached to the System, which controls their execution through a particular interpretation strategy, which is guided by the semantic network. The interpretation process carried out by InterIMAGE has two sequential, complementary phases. The first phase – the top-down step – is model-driven and generates hypotheses of the occurrence of image objects on the scene. The final instance network results from the data-driven, bottom-up analysis, responsible from judging the object hypotheses, validating or discarding them based on user defined decision rules. This paper focuses on describing the capabilities recently introduced in InterIMAGE, such as: multitemporal interpretation; debugging functionality; feature extraction and exploration; the graphical interface for creation of decision rules; and the System's basic suite of image processing operators. In order to make a comprehensive description of these capabilities some simple examples will be also presented.

1. INTRODUCTION

Traditionally, image interpretation has been treated as a pattern recognition problem. In (Jain et al., 2000) pattern recognition techniques are grouped in categories, some of which are particularly important for the analysis of remote sensing data: statistical methods; methods based in machine learning; and structural methods.

Statistical methods (Webb, 2002) can be regarded as the conventional solution in terms of remote sensing. In this category, object classes are represented by probability density functions defined over a predetermined attribute space. Appropriately modeling probability densities is the main challenge of this type of approach. Limitations as to the number of the available training patterns usually imply in the adoption of simple models, and in poor classification performance.

Methods based on machine learning (Li et al., 2000; Mciver et al., 2001; Zhong et al., 2008; Chi and Ersoy, 2005), are directed towards the learning of complex relations among sample patterns, even in the absence of explicit models. The main disadvantage of these methods is their high demand for training samples, what excludes them as an alternative for many remote sensing applications.

Structural methods involve complex patterns. They usually adopt a hierarchical approach in which descriptions of the patterns are based on simpler patterns recursively, until primitive patterns are reached. Particularly relevant in this

context are the so-called knowledge-based or cognitive systems (Sagerer and Niemann, 1997; Liedtke et al., 1999; Bückner et al., 2001; Schiewe et al., 2001; Centeno et al., 2003). The main focus of those systems is the modeling of the classes of objects expected to be found in an image through the explicit representation of prior knowledge about their spectral, morphological or topological characteristics. Such knowledge, acquired from a human specialist, can reduce significantly the demand for training patterns.

There are several other advantages of embodying image understanding knowledge into explicit structures (Crevier and Lapage, 1997). First, knowledge can be easily added to a knowledge base, without modifying preexisting rules. When laid out explicitly, knowledge can be more easily validated, since contradictions and omissions become apparent. Knowledge structures also favor interactive problem solution, providing a way to explore alternative means of extracting information from images. Last but not least, explicit represented knowledge provides for easier collaboration, for knowledge interchange among those tackling similar problems.

The InterIMAGE Project aims at developing an open-source, knowledge-based framework for automatic interpretation of remote sensing data. The Project, which began in 2008, is leaded by the Computer Vision Lab of the Catholic University of Rio de Janeiro in collaboration with the Brazilian National Institute for Space Research (INPE) and the TNT Institute of the Hannover Leibniz University.

In this paper we present the main features of the InterIMAGE system. Special attention is given to describing the functionalities recently added to the system, such as: multitemporal interpretation support; knowledge model debugging functionality; feature extraction and exploration; the graphical interface for creation of decision rules; and the System's basic suite of image processing operators. In order to make a comprehensive description of these capabilities, some simple examples are also presented.

In the remainder of this paper we describe the basic characteristics of InterIMAGE and the interpretation strategy implemented by the system (Section 2). In Section 3 we describe the capabilities recently introduced in InterIMAGE. Finally, some conclusions are presented in Section 4.

2. SYSTEM DESCRIPTION

InterIMAGE is a multi-platform framework, counting currently with implementations for LINUX and Windows operational systems. The System is coded in C++, using also the QT4 cross-platform application development framework (Summerfield, 2010), the Visualization Toolkit (VTK) class library (Schroeder et al., 2006) and Terralib (Camara, 2000), a GIS classes and functions library developed at INPE.

InterIMAGE provides support for the integration of image processing operators in the interpretation process and, as such operators are treated as external programs by its control mechanism, they can be coded in any computer language, and can even be proprietary programs. The InterIMAGE framework offers, nonetheless, a suit of basic operators (http://www.dpi.inpe.br/terraaida), assembled with the classes and functions supplied by the TerraLib library (Câmara et al., 2000).

InterIMAGE is founded on GeoAIDA (Bükner et al., 2001), developed at the TNT Institute (Institut für Informationsverarbeitung) at the Leibniz Hannover University, Germany, and it inherited from that system its interpretation engine and some of its basic knowledge representation structures.

The central goal of InterIMAGE is to aid the user in the interpretation of a scene, providing thus a symbolic meaning to the different regions that compose the scene. The reader should note that the term scene here refers to a geographic region and not to one particular remote sensing image. In fact, for the interpretation of a scene with InterIMAGE, one can integrate different georreferenced data covering the region of interest, such as a collection of images acquired at different points in time or acquired by different sensors, or even data from a vector, GIS database.

InterIMAGE's interpretation process is guided by a knowledge model, which embodies and explicitly represents prior knowledge about a particular geographic region and about the objects expected to be found in it. The knowledge model can correspond to the user's prior knowledge or the knowledge acquired by analyzing the input data (RS images or GIS layers) through the tools provided by system.

The basic element of a knowledge model in InterIMAGE is a semantic network (fig. 1). Each node of the network represents a concept – a class of objects expected to be found in the scene.
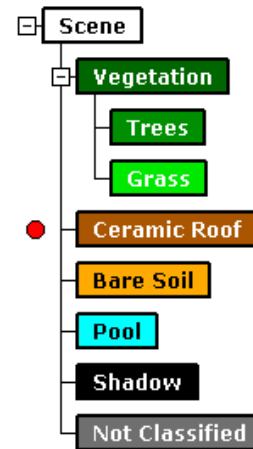

Figure 1. A Semantic Network in InterIMAGE

The user can define for each network node specific operations – which can process input data with particular characteristics, e.g. different input images – to search for the objects of the class associated to the node.

During interpretation, InterIMAGE's control process will visit the nodes of the semantic network in a particular order, executing the operations associated to each node. Initially, the control process will visit the nodes from top to bottom – from the root to the leaf nodes – in the so called *top-down step* of the interpretation process. After reaching the leaf nodes, the interpretation process will traverse the network in the opposite direction, in the so called *bottom-up step*.

The operations associated to each node are specialized for the top-down or bottom-up interpretation steps. *Top-down operations* are responsible for what is usually regarded in the remote sensing field as *feature extraction*, which in this context means finding regions on the scene associated to the occurrence of a class of objects. Hence, the output of a top-down operation is a set of *object hypotheses*.

Why the term *hypothesis* was used in the last paragraph? This should become clearer later, but the basic idea is that the results of top-down operations need to be validated (or discarded) in the subsequent, bottom-up interpretation step. Top-down operations may be though of as local processes linked to the nodes of the semantic network. As those processes do not communicate with each other directly, they may generate results that are inconsistent among one another, i.e. they may associate to the same geographic region different classes of objects. It is the task of the *bottom-up operations* to resolve the conflicts among the hypotheses generated in the top-down step, i.e. to decide which hypotheses will prevail. The hypotheses validated through the bottom-up operations are regarded as *object instances*.

Bottom-up and top-down operations are generally composed by two elements, an operator (top-down or bottom-up) and a decision rule. In practical terms, the user can select an operator and set its parameters, and define decision rules at each node of

the semantic network, for the top-down and for the bottom-up interpretation steps. In doing so, the user defines how to find the possible occurrences of a specific concept, and how to choose among the concurrent occurrences of the different concepts.

An *operator* (top-down or bottom-up) is an executable program – external to the System's core – called by the control mechanism during interpretation. An operator, therefore, embodies implicit knowledge about one or more concepts. The term *implicit* has to do with the fact that the problem solving knowledge encoded in the operator cannot be manipulated or altered by the user of InterIMAGE, which can merely set the parameters of those operators through the system's GUI.

The user knows what the operator does through its documentation, but he/she usually will not necessarily know precisely how it does it. InterIMAGE is very flexible with respect to the integration of new operators. The expert user can create a new operator by encoding a specific algorithm into an executable program, which can be coded in any programming language. This program will need, however, to conform to a communication protocol (specified in InterIMAGE's user guide), so that that the System will know how to call the program, set its parameters, and process its outcome.

Top-down operators are responsible for partitioning the scene into sub-regions, which are the geographic representation of the object hypotheses. *Bottom-up operators* are responsible for evaluating the object hypotheses – validating, discarding or altering them.

After the execution of an operator (top-down or bottom-up), InterIMAGE will execute user defined decision rules, which can be designed through a specific visual language (Section 3.2). Their basic tasks are to filter, group or reclassify the results of the operators according to a particular criteria. A bottom-up decision rule can also contain commands to resolve eventual spatial conflicts among the hypotheses generated in the top-down step. A decision rule can be though of a set of actions that complement what an operator does, as it may alter or refine the operator's output.

The output of the interpretation process is a symbolic description of the scene, consisting of an instance network – a network of object instances. Based on the instance network, the system is able to create different thematic maps representing the different levels of concepts in the semantic network.

3. INTERIMAGE ENHANCEMENTS

In the last few years – since the report presented in (Oliveira et al., 2008), in the GEOBIA2008 Conference – new functionality has been added to InterIMAGE, and in the remainder of this paper we report on the most relevant of those new functions.

3.1 Basic Operator Suit

Although, as it has been said in Section 2, the user can add its own classification or feature extraction methods to the system as external operators, the standard version of InterIMAGE comes with a suit of basic, generic operators. This suit of operators was constructed using primarily the functions and methods present in the TerraLib library, and for that reason they are called

TerraAida operators. The source code of those operators, as well as binaries and documentation can be found at http://www.dpi.inpe.br/terraaida.

We present below a list of the currently available TerraAIDA operators.

- *TerraAIDA Arithmetic*: generates segments based on the thresholding of a grayscale image resultant from a user defined arithmetic operation over the bands of an input image.

- *TerraAIDA Baatz&Schäpe Segmentation*: performs a segmentation of an input image using the algorithm proposed in (Baatz and Shäpe, 2000).

- *TerraAIDA Checkerboard Segmentation*: creates rectangular shaped image segments according to user defined specifications.

- *TerraAIDA NDVI Segmentation*: generates segments based on the thresholding of a grayscale image whose digital numbers are equivalent to the Normalized Difference Vegetation Index (NDVI) (Sellers, 1985).

- *TerraAIDA Region Growing Segmentation*: performs a segmentation of an input image through the algorithm proposed in (Bins et al., 1996).

- *TerraAIDA ShapeFile Import*: imports polygons from a shapefile (ESRI, 1998) and generates segments based on the imported polygons.

- *TerraAIDA ShapeFile Intersection*: creates segments from the intersection of the regions associated to higher level hypotheses (from the father, semantic network node) and a shapefile provided by the user.

- *TerraAIDA Decision Tree Classifier*: classifies regions based on the supervised C4.5 algorithm (Quinlan, 1993). This operator works together with one of the segmentation operators listed above, classifying the resulting segments, based on samples selected by the user.

The operators listed above are to be used in the top-down interpretation step (Section 2). They are, therefore, used in the process of generating object hypotheses of the concepts associated to the semantic network nodes they are attached to. The user should note that all but the last listed operator provide solely segments, that is, they do not perform any classification of the output segments. The user can, however, define decision rules (see next section) that will actually classify the segments, selecting only the ones that are to be considered hypotheses of the corresponding semantic network nodes, and assigning to those hypotheses membership values that can later be used on the process of deciding among overlapping hypotheses.

3.2 Decision Rules

Decision rules can be used either to classify image segments generated by one of the top-down, segmentation operators listed in the prior section, or to decide among the concurrent object hypotheses, in the bottom-up interpretation step.

A graphical user interface was devised to aid the user in the definition of decision rules. The elements of this interface are depicted in fig. 2.
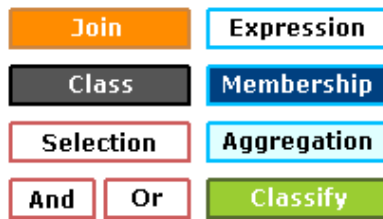

Figure 2. Basic Decision Rule Elements

The *Class* element invokes all hypotheses of a specific object class (associated to a node of a semantic network). The *Join* element joins hypotheses associated to different object classes.

In a decision rule, InterIMAGE can calculate a number of different attributes of a object hypothesis, attributes based on spectral values, shape, texture and topological characteristics of the image segments associated to those hypotheses. These attributes can be used in the *Selection* of hypotheses, based on a crisp threshold. Selection rules can be combined by logical *And* or *Or* operations.

The *Expression* element enables the user to create named variables form the object hypotheses properties. The values of those variables, are stored within the object hypotheses, and can be used later in the same decision rule, or in a decision rule associated to another semantic network node. The *Membership* element enables the user to assign membership values to the object hypotheses, which can be done through a combination of membership functions such as shown in fig. 3. The *Aggregation* element allows for the aggregation of attribute values over a set of object hypotheses, and the *Classify* element – enabled only in the bottom-up interpretation step, i.e., in a bottom-up decision rule – is used to resolve spatial conflicts among competing hypotheses.
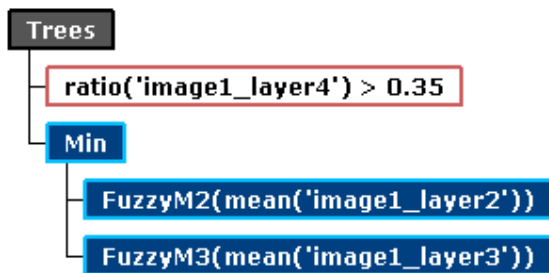

Figure 3. A Top-Down Decision Rule Example

Figure 3 shows a simple top-down decision rule example, for the *Trees* node in the semantic network shown in fig.1. In this case a segmentation top-down algorithm was attached to the *Trees* node, and all the segments generated by the operator are submitted to the rule. First, all the segments for which the ratio of the mean values of band 4 (with respect to all other bands of *image1*) are higher than a certain crisp threshold are selected to be considered as *Trees* hypotheses. Than, those hypotheses will receive a membership value that is the minimum value of the

FuzzyM2 and *FuzzyM3* membership functions, defined, respectively over the mean pixel values of bands 2 and 3 of *image1*. The membership functions are defined interactively by the user, function *FuzzyM2* is shown in fig.4.
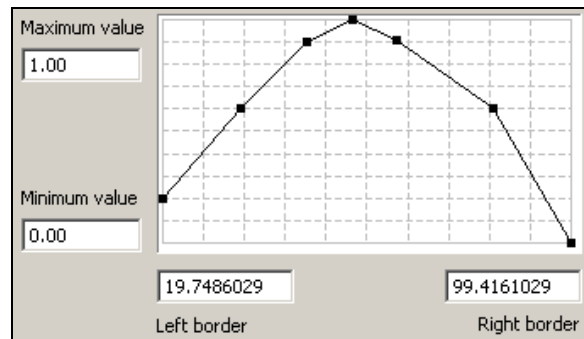

Figure 4. A Membership Function Example

Figure 5 shows a simple bottom-up decision rule example, for the *Vegetation* node in the semantic network shown in fig.1. Basically what the rule does is to select all the hypotheses generated in the top-down interpretation step, for the nodes *Trees* and *Grass*, and perform an operation that will decide on the wining hypotheses based on their membership values. It is important to note that if there is a partial spatial overlap between a hypothesis of *Trees* and a hypothesis of *Grass*, the hypothesis with the lower membership value will not be completely discarded, only the overlapping part of it will be eliminated, i.e. the region associated to it will become smaller.

Figure 5. A Bottom-Up Decision Rule Example

The last operation of a decision rule is a merging operation; this operation is responsible for grouping, spatially, the resulting hypotheses of a decision rule. There are three possibilities: *global merge* – all the hypotheses are combined into a single hypothesis, which can cover a non-contiguous area; *merge connected* – each group of spatially connected hypotheses will be combined into a single hypotheses, which will cover a contiguous area; or *no merge*, the hypotheses that resulted from the decision rule will not be merged.

3.3 Multitemporal Interpretation Support

InterIMAGE provides support for multitemporal interpretation by offering alternative control strategies that take into consideration the temporal dimension of the interpretation problem. And this temporal dimension is made explicit in the semantic network through the so-called *temporal nodes*. The nodes *2000* and *2001* in the network shown in fig. 6 are temporal nodes. Each network branch underneath a temporal node is regarded as a *temporal branch*.

InterIMAGE implements two alternative interpretation strategies, namely: *sequential* and *synchronous multitemporal strategies*.

In the *sequential interpretation strategy*, the interpretation process will stop, in the top-down step, when it reaches the temporal nodes. Then it will proceed just for the first temporal branch. When processing (top-down and bottom-up) is complete for the first temporal branch, the validated hypotheses of the children of the first temporal node will be use to generate initial hypotheses for the children of the second temporal node, and this process will continue until the last temporal branch is processed, when the interpretation will resume for the node that is the father of the temporal nodes.

It is important to observe that the children of the temporal nodes need not correspond to similar concepts, and that the temporal nodes can be placed at any hierarchical level of the semantic network.
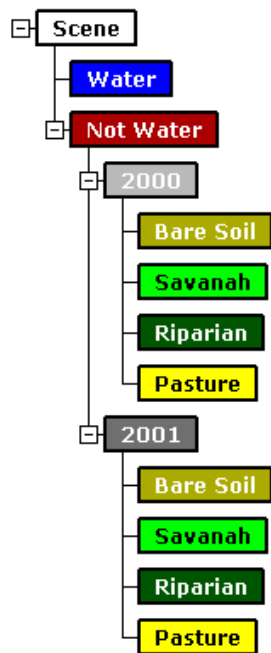

Figure 6. Semantic Network with Temporal Branches

In the *synchronous interpretation strategy*, interpretation will proceed as normal during the top-down step, i.e. the temporal branches will be processed in parallel. In the bottom-up step, however, interpretation will halt at each temporal branch until each branch is completely processed (top-down and bottom up). Then the membership values of the competing hypotheses of the children of the temporal nodes will be subjected to an operation called *multitemporal merge*, which will use class transition possibility values – expressed in a fuzzy transition matrix – to update the hypotheses membership values, as reported in the work (Feitosa et al., 2009). These updated membership values will in turn be used in deciding which hypotheses will prevail.

3.4 Interactivity Support

A number of capabilities have been also included in InterIMAGE to allow the definition of knowledge models in an interactive fashion and to permit the debugging of such models.

In the semantic network depicted in fig. 1. there is a red circle next to the *Ceramic Roof* node. It indicates a *break point* in the interpretation process, and signals to that process that when it reaches that node it should stop and enter the *stepwise execution mode*. In such mode, the user can command the process to proceed to the next node, to resume the execution as normal, or he/she can invoke the *analysis manager*.

The analysis manager has a graphical interface that allows the user to experiment with different operators and decision rules, by modifying such elements – i.e. trying out different segmentation operators or segmentation parameters, and alternative decision rules – and visually investigating their results.

The analysis manager also allows the user to spatialize spectral, textural, morphological and topological attributes over the image segments that represent the object hypotheses, and to produce histograms and scatter plots from their values.

In the stepwise execution mode, the user can also *undo* the operations performed at a node of the semantic network.

Another important feature is the possibility to visualize intermediate results of the interpretation process. The user can select any result of any processing step, before or prior to the execution of top-down and bottom-up operations, so that he/she will be able to investigate the effects of such operations to any hypothesis generated during the interpretation process

There are many other features which will not be described here due to the lack of space, such as de interface for creating and editing shapefiles, definition of regions of interest (ROIs) and the definition of samples for supervised classification operators.

Further details about the system, including technical and user documentation, source code and executable versions are available at the InterIMAGE Project's website[1], at the InterIMAGE wiki[2] and at the TerraAIDA website[3].

4. CONCLUSION

In this paper the basic characteristic and interpretation process implemented by the InterIMAGE System, an open source knowledge-based framework for automatic interpretation of remote sensing data developed by the Computer Vision Lab of the Catholic University of Rio de Janeiro in collaboration with the Brazilian National Institute for Space Research (INPE) and with the TNT Institute of the Leibniz Hannover University have been presented.

We believe that with the recently included functionality, InterIMAGE has become an interesting alternative to commercial software packages for the interpretation of remote sensing image data. The most obvious advantage being that the system can be acquired for free.

Another important advantage with respect to the commercial packages is that InterIMAGE is not a black-box. Besides it

[1] http://www.lvc.ele.puc-rio.br/projects/interimage
[2] http://wiki.dpi.inpe.br/doku.php?id=interimage
[3] http://www.dpi.inpe.br/terraaida

documentation, the user with some programming skills can download and investigate its source code, and find out exactly how the System's functions work.

Moreover, users can add to InterIMAGE their own methods in the form of operators, and can even modify some functionalities of the system. Being that so, we believe that InterIMAGE can be used as a software framework for the development of scientific research in the fields of Object Based Image Analysis and Computer Vision.

AKNOWLEGMENTS

REFERENCES

Ain, K., Duin R., Mao, J., 2000. Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 22 (1), pp. 4-37.

Baatz, M., Schape, A., 2000. Multiresolution segmentation: an optimization approach for high quality multi-scale image segmentation. In: *XII Angewandte Geographische Informationsverarbeitung*, Wichmann-Verlag, Heidelberg.

Bins, L. S., Fonseca, L. M. G., Erthal, G. J., Ii, F. M., 1996. Satellite imagery segmentation: a region growing approach. In: *VIII Simpósio Brasileiro de Sensoriamento Remoto*, Salvador, BA. pp. 677-680.

Bückner, J., Pahl, M., Stahlhut, O., Liedtke, C.-E., 2001. GEOAIDA - A knowledge-based automatic image data analyzer for remote sensing data. In: *ICSC Congress On Computational Intelligence Methods And Applications 2001 - CIMA 2001*, Bangor, Wales, UK. Proceedings of the Congress on Computational Intelligence Methods and Applications 2001 - CIMA 2001 (CD-ROM).

Camara, G., Souza, R. M., Pedrosa, B. M., Vinhas, L., Monteiro, A. M., Paiva, J. C., Carvalho, M. T., Gattass, M., 2000. TerraLib: Technology in support of GIS innovation. In: *Workshop Brasileiro de GeoInformática*, São Paulo, Brasil, pp. 126-133.

Centeno, J. A. S., Antunes, A. F. B., Trevizan, S.; Correa, F., 2003. Mapeamento de áreas permeáveis usando uma metodologia orientada a regiões e imagens de alta resolução. *Revista Brasileira de Cartografia*, 55 (1), pp. 48-56.

Chi, H. M., Ersoy, O. K., 2005. A statistical self-organizing learning system for remote sensing classification. *IEEE Transactions on Geoscience and Remote Sensing*, 43, pp.1890-1900.

Crevier, D., Lepage, R, 1997. Knowledge-Based Image Understanding Systems: A Survey. *Computer Vision and Image Understanding*, 67 (2), pp. 61-185.

ESRI, 1998. *Shapefile Technical Description - ESRI White Paper*, July 1998.

Feitosa, R. Q., Costa, G. A. O. P., Mota, G. L. A., Pakzad, K., Costa, M. C. O., 2008. Cascade Multitemporal Classification based on Fuzzy Markov Chains. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64 (2), pp. 159-170.
Li, D., Di, K., Li, D., 2000. Land Use Classification of Remote Sensing Image with GIS on Spatial Data Mining Tehcniques. *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXIII, Part B3, pp. 238-245.

Liedtke, C.-E., Bückner, J., Grau, O., Growe, S., Tönjes, R., 1999. AIDA: A system for the knowledge-based interpretation of remote sensing data. In: *Third International Airborne Remote Sensing Conference And Exhibition*, 1999, Copenhagen, Denmark, Proceedings of the Third International Airborne Remote Sensing Conference and Exhibition.

Mciver, D. K., Friedl, M. A., 2001. Estimating pixel-scale land cover classification confidence using nonparametric machine learning methods. *IEEE Transactions on Geoscience and Remote Sensing*, 39 (9), pp. 1959-1968.

Oliveira, D. A. B., COSTA, G. A. O. P., Feitosa, R. Q., Castejon, E. F., Fonseca, L. M. G., 2008. InterIMAGE: An Open Source, Knowledge Based Framework for Automatic Interpretation of Remote Sensing Data. In: *GEOBIA, 2008 - Geographic Object Based Image Analysis for the 21st Century*, 2008, Calgary.

Quinlan, J. R., 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

Sagerer, G., Niemann, H., 1997. Semantic networks for understanding scenes. *Advances in Computer Vision and Machine Intelligence.* 1. ed., New York: Plenum Publishing Corporation, 500p.

Schiewe, J., Tufte, L., Ehlers, M., 2001. Potential and problems of multi-scale segmentation methods in remote sensing. *Geo-Informations-Systeme*, 6, pp. 34-39.

Schroeder, W., Martin, K., Lorensen, B., 2006. *The Visualization Toolkit: An Object-Oriented Approach To 3D Graphics*, 4th Edition, Kitware.

Sellers, P. J., 1985. Canopy Reflectance, Photosynthesis and Transpiration. *International Journal of Remote Sensing*, 6, pp. 1335-1372.

Summerfield, M., 2010. *Advanced Qt Programming: Creating Great Software with C++ and Qt 4*, Addison-Wesley, 500p.

Webb, A. R., 2002. *Statistical Pattern Recognition*, Second Edition, Wiley.

Zhong, P., Zhang, P., Wang, R. 2008. Dynamic Learning of SMLR for Feature Selection and Classification of Hyperspectral Data. *IEEE Geoscience and Remote Sensing Letters*, 5 (2), pp. 280-284.