Efficient Multi-Band Texture Analysis for Remotely Sensed Data Interpretation in Urban Areas

Javier Plaza, Antonio Plaza
Department of Computer Science, University of Extremadura
Avda. de la Universidad s/n, E-10071 Caceres, Spain
Email: {jplaza, aplaza}@unex.es

Paolo Gamba, Giovanna Trianni
Department of Electronics, University of Pavia
Via Ferrata, 1, 27100 Pavia, Italy
Email: {paolo.gamba, giovanna.trianni}@unipv.it

Abstract— **Texture analysis is a long-standing and important problem in image-based urban characterization. A variety of approaches and methods have been proposed in the past to deal with urban texture segmentation and classification. However, texture characterization is particularly complex when the image data is composed of several spectral bands at different wavelengths, as in the case of remotely sensed hyperspectral images, in which hundreds of spectral bands are often available. Such images have two domains which can be analyzed: the spectral domain and the spatial domain. In this paper, we develop a joint spatial/spectral classification approach for hyperspectral imagery which is shown to perform effectively in highly complex urban environments. Experimental results are provided using a hyperspectral scene with extensive ground-truth, collected over the town of Pavia in Italy. To address the high computational requirements of the algorithm, we also develop a parallel implementation which is tested in this work using a massively parallel supercomputer at NASA's Goddard Space Flight Center in Maryland.**

I. Introduction

The integration of spatial and spectral responses in hyperspectral image analysis has been identified as a highly desirable objective by the remote sensing community, in particular for urban data analyses in which the high spectral dimensionality of the data is often complemented by very high spatial resolution [1], [2]. The need for joint spatial/spectral approaches results from shortcomings of available data processing techniques. For instance, previous research has demonstrated that the high-dimensional data space spanned by hyperspectral data sets is usually empty, indicating that the data structure involved exists primarily in a subspace [3]. A commonly used approach to reduce the dimensionality of the data is the principal component transform (PCT). However, this approach is characterized by its global nature and cannot preserve subtle spectral differences required to obtain a good discrimination of classes [4], [5]. Further, this approach relies on spectral properties of the data alone, thus neglecting the information related to the spatial arrangement of the pixels in the scene. As a result, there is a need for feature extraction techniques able to integrate the spatial and spectral information available from the data simultaneously [6]. In this contex, texture information may assist in accurately characterizing the data, although it is worth noting that most available techniques to characterize multi-band texture are based on the consideration of spectral information separately from spatial information, and thus the two types of information are not treated simultaneously. By taking into account the complementary nature of spatial and spectral information in simultaneous fashion, it may be possible to alleviate the problems related to each of them taken separately and improve segmentation and classification results in urban analysis scenarios.

While such integrated spatial/spectral developments hold great promise in the field of remote sensing data analysis, they introduce new processing challenges [7]. In particular, there is a need for fast response in many hyperspectral imaging oriented applications. For instance, real-time response is required in time-critical studies such as detection and monitoring of fires in urban areas, or in target detection for military purposes. The concept of Beowulf cluster was developed, in part, to address such challenges. The goal was to create parallel computing systems from commodity components to satisfy specific requirements for the Earth and space sciences community.

To address the need for cost-effective and innovative algorithms in this emerging new area, this paper develops a novel three-stage process for computationally efficient classification of hyperspectral urban imagery. The process consists of the following stages:

1) First, we quantize the spatial and the spectral information contained in the hyperspectral urban data set (in simultaneous fashion) by using mathematical morphology [8] concepts. Morphology is a consolidated, spatial-based image processing technique which has been shown to be successful for characterization of urban areas. In this work, we develop several vector-ordering strategies to extend mathematical to multi-dimensional imagery [9], and further propose a set of extended morphological texture features, obtained from increasing series of morphological opening and closing operations.
2) Second, we construct a training set based on morphological features and use it to train a multi-layer perceptron (MLP) neural network architecture which is effectively trained [10] to discriminate among the considered urban classes.

The remainder of the paper is organized as follows. Section II describes the proposed methodology for morphological/neural classification. In order to account for the heavy computational load introduced by the proposed multi-band texture analysis approach in hyperspectral urban analysis en-

1-4244-0712-5/07/$20.00 ©2007 IEEE.

vironments, Section III develops parallel processing support for the two stages of the proposed algorithm. In section IV the proposed approach is quantitatively and comparatively assessed in the context of urban mapping applications by drawing comparisons to other standard classification approaches, using an urban data set (with extensive ground-truth) collected over the town of Pavia, Italy, by the Digital Airborne Imaging Spectrometer (DAIS 7915) in the framework of the EU HySens campaign led by DLR. Performance data are measured in a massively parallel Beowulf cluster called Thunderhead and available at NASA's Goddard Space Flight Center in Maryland.

II. Morphological/Neural Classification

In this section we describe a new methodology for the classification of urban hyperspectral images. First, we briefly introduce the similarity metric used in this work to perform spectral matching. Then, a morphological feature extraction algorithm based is presented. Finally, we describe the MLP classifier used in this work.

A. Spectral similarity metric

Let us first denote by f a hyperspectral image defined on an N-dimensional (N-D) space, where N is the number of channels or spectral bands in the image. A widely used technique to measure the similarity between spectral signatures in the input data is the spectral angle mapper (SAM) [11], which can be used to measure the spectral similarity between two pixel vectors, $f(x,y)$ and $f(i,j)$, i.e., two N-D vectors at discrete spatial coordinates (x,y) and $(i,j) \in Z^2$, as follows:

$$\text{SAM}(f(x,y), f(i,j)) = cos^{-1} \frac{f(x,y) \cdot f(i,j)}{\|f(x,y)\| \cdot \|f(i,j)\|} \tag{1}$$

B. Morphological feature extraction algorithm

The proposed feature extraction method is based on mathematical morphology [8] and spectral matching concepts. The goal is to impose an ordering relation (in terms of spectral purity) in the set of pixel vectors lying within a spatial search window (called structuring element) designed by B. This is done by defining a cumulative distance between a pixel vector $f(x,y)$ and all the pixel vectors in the spatial neighborhood given by B (B-neighborhood) as follows [9]: $D_B[f(x,y)] = \sum_i \sum_j \text{SAM}[f(x,y), f(i,j)]$, where (x,y) refers to spatial coordinates in the B-neighborhood. From the above definitions, two standard morphological operations called erosion and dilation can be respectively defined as follows:

$$(f \otimes B)(x,y) = argmin_{(s,t) \in Z^2(B)} \sum_s \sum_t \text{SAM}(f(x,y), f(x+s, y+t)) \tag{2}$$

$$(f \oplus B)(x,y) = argmax_{(s,t) \in Z^2(B)} \sum_s \sum_t \text{SAM}(f(x,y), f(x-s, y-t)) \tag{3}$$

Using the above operations, the opening filter is defined as an erosion followed by a dilation:

$$(f \circ B)(x,y) = [(f \otimes C) \oplus B](x,y) \tag{4}$$

On the other hand, the closing filter is defined as a dilation follwed by an erosion:

$$(f \bullet B)(x,y) = [(f \oplus C) \otimes B](x,y) \tag{5}$$

The composition of opening and closing operations is called a spatial/spectral profile [6], which is defined as a vector which stores the relative spectral variation for every step of an increasing series. Let us denote by $\{(f \circ B)^\lambda(x,y)\}, \lambda = \{0, 1, ..., k\}$, the *opening series* at $f(x,y)$, meaning that several consecutive opening filters are applied using the same window B. Similarly, let us denote by $\{(f \bullet B)^\lambda(x,y)\}, \lambda = \{0, 1, ..., k\}$, the *closing series* at $f(x,y)$. Then, the spatial/spectral profile at $f(x,y)$ is given by the following vector:

$$\begin{aligned} p(x,y) = &\{\text{SAM}((f \circ B)^\lambda(x,y), (f \circ B)^{\lambda-1}(x,y))\} \\ &\cup \{\text{SAM}((f \bullet B)^\lambda(x,y), (f \bullet B)^{\lambda-1}(x,y))\} \end{aligned} \tag{6}$$

Here, the step of the opening/closing series iteration at which the spatial/spectral profile provides a maximum value gives an intuitive idea of both the spectral and spatial distribution in the B-neighborhood [6]. As a result, the profile can be used as a feature vector on which the classification is performed using a spatial/spectral criterion.

C. Multilayer perceptron classifier

In this section we describe a supervised classifier based on a multi-layer perceptron (MLP) neural network with back-propagation learning, which is trained with the spatial/spectral features resulting from the morphological feature extraction algorithms. This neural architecture has been shown in previous work to be robust for classification of hyperspectral imagery [10], but the use of morphological features as input features for the classification represents a novel contribution.

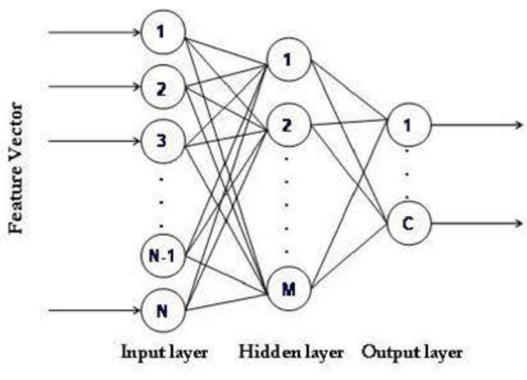

Fig. 1. MLP neural network topology.

The architecture adopted for the proposed MLP-based neural network classifier is shown in Fig. 1. The number of input neurons equals the number of spectral bands acquired by the sensor. However, in the case of PCT-based pre-processing commonly adopted in hyperspectral analysis or our proposed morphological feature extraction, the number of neurons at the input layer equals the dimensionality of feature vectors used for classification purposes. The second layer is the hidden layer, in which the number of nodes, M, is usually estimated empirically. Finally, the number of neurons at the output layer, C, equals the number of distinct classes to be identified in the input data. With the above architecture in mind, the standard back-propagation learning algorithm used by the MLP neural architecture in Fig. 1 can be outlined by the following steps:

1) *Forward phase*. Let the individual components of an input pattern be denoted by $f_j(x,y)$, with $j = 1, 2, ..., N$. The output of the neurons at the hidden layer are obtained as: $H_i = \varphi(\sum_{j=1}^{N} \omega_{ij} \cdot f_j(x,y))$ with $i = 1, 2, ..., M$, where $\varphi(\cdot)$ is the activation function and ω_{ij} is the weight associated to the connection between the i-th input node and the j-th hidden node. The outputs of the MLP are obtained using $O_k = \varphi(\sum_{i=1}^{M} \omega_{ki} \cdot H_i)$, with $k = 1, 2, ..., C$. Here, ω_{ki} is the weight associated to the connection between the i-th hidden node and the k-th output node.
2) *Error back-propagation*. In this stage, the differences between the desired and obtained network outputs are calculated and back-propagated. The *delta* terms for every node in the output layer are calculated using $\delta_k^o = (O_k - d_k) \cdot \varphi^{'}(\cdot)$, with $i = 1, 2, ..., C$. Here, $\varphi^{'}(\cdot)$ is the first derivative of the activation function. Similarly, *delta* terms for the hidden nodes are obtained using $\delta_i^h = \sum_{k=1}^{C}(\omega_{ki} \cdot \delta_i^o) \cdot \varphi(\cdot))$, with $i = 1, 2, ..., M$.
3) *Weight update*. After the back-propagation step, all the weights of the network need to be updated according to the *delta* terms and to η, a learning rate parameter. This is done using $\omega_{ij} = \omega_{ij} + \eta \cdot \delta_i^h \cdot f_j(x,y)$ and $\omega_{ki} = \omega_{ki} + \eta \cdot \delta_k^o \cdot H_i$. Once this stage is accomplished, another training pattern is presented to the network and the procedure is repeated for all incoming training patterns.

Once the back-propagation learning algorithm is finalized, a classification stage follows, in which each input pixel vector is classified using the weights is obtained by the network during the training stage [10]. In this work, training of the neural network is performed by selecting a random set of pixels from the known ground-truth of the data. One of our future research lines is directed towards the automatic selection of the most useful training patterns for classification purposes.

III. Parallel implementation

This section develops a parallel implementation of the proposed morphological/neural classification technique which has been specifically optimized for execution on massively parallel, Beowulf-type commodity clusters. First, we describe the parallel morphological feature extraction framework addressing the impact of communications in the parallel architecture. The section concludes with an overview of the proposed parallel framework for efficient execution of the MLP-based algorithm with back-propagation learning.

A. Parallel morphological feature extraction

Two types of partitioning can be exploited in the parallelization of spatial/spectral algorithms such as the morphological feature extraction technique presented above [7]. In this work, we adopt a spatial-domain partitioning approach due to several reasons. First, the application of spatial-domain partitioning is a natural approach for morphological image processing, as many operations require the same function to be applied to a small set of elements around each data element present in the image data structure, as indicated in the previous subsection. A second reason has to do with the cost of inter-processor communication. In spectral-domain partitioning, the window (structuring element)-based calculations made for each hyperspectral pixel need to originate from several processing elements, in particular, when such elements are located at the border of the local data partitions (see Fig. 2), thus requiring intensive inter-processor communication.

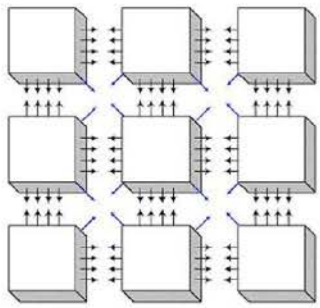

Fig. 2. Communication framework for the morphological feature extraction algorithm.

To address the above issues, we have developed a data replication-based strategy in which border data is replicated rather than communicated between adjacent processors. This strategy results in higher computational performance as shown in previous work [7]. The inputs to the parallel algorithm are an N-dimensional hyperspectral data cube f and a structuring element B. The output is a set of morphological profiles for each pixel. A pseudo-code of the parallel algorithm is given below:

1) Obtain information about the parallel system, including the number of processors, P, and each processors identification number, $\{p_i\}_{i=1}^{P}$.
2) Using B and the information obtained in Step 1, determine the total volume of information, R, that needs to be replicated from the original data volume, V, according to the data communication strategies outlined above, and let the total workload W to be handled by the algorithm be given by $W = V + R$.
3) Set $\alpha_i = P/w_i$ for all $i \in \{1, ..., P\}$, where w_i is assumed to be the speed of all processors in the homogeneous parallel cluster.

4) For $m = \sum_{i=1}^{P} \alpha_i$ to $(V + R)$, find $k \in \{1, .., P\}$ so that $w_k \cdot (\alpha_k + 1) = \min\{w_i \cdot (\alpha_i + 1)\}_{i=1}^{P}$ and set $\alpha_k = \alpha_k + 1$.
5) Use the resulting $\{\alpha_i\}_{i=1}^{P}$ to obtain a set of P spatial-domain partitions (with overlap borders) of W, and send each partition to processor p_i, along with B.
6) Calculate the morphological profiles $p(x, y)$ for the pixels in the local data partitions (in parallel) at each processor in the parallel system.
7) Collect all the individual results and merge them together to produce the final output.

B. Parallel multi-layer perceptron classifier

The parallel MLP classifier developed in this work is based on a hybrid partitioning scheme, in which the hidden layer is partitioned using neuronal level parallelism and weight connections are partitioned on the basis of synaptic level parallelism [12]. As a result, the input and output neurons are common to all processors, while the hidden layer is partitioned so that each heterogeneous processor receives a number of hidden neurons which depends on its relative speed. Each processor stores the weight connections between the neurons local to the processor. Since the fully connected MLP network is partitioned into P partitions and then mapped onto P heterogeneous processors using the above framework, each processor is required to communicate with every other processor to simulate the complete network. For this purpose, each of the processors in the network executes the three phases of the back-propagation learning algorithm described above.

The inputs to the parallel MLP algorithm are an N-dimensional image cube f and a set of training patterns $f_j(x, y)$. The output is a set of classification labels for each image pixel. The algorithm can be summarized by the following steps:

1) Obtain the number of nodes present in the cluster architecture P which will be used to obtain a set of P partitions of the hidden layer and map the resulting partitions among the P processors (which also store the full input and output layers along with all connections involving local neurons).
2) *Parallel training*. For each considered training pattern, the following three parallel steps are executed:
   a) *Parallel forward phase*. In this phase, the activation value of the hidden neurons local to the processors are calculated. For each input pattern, the activation value for the hidden neurons is calculated using $H_i^P = \varphi(\sum_{j=1}^{N} \omega_{ij} \cdot f_j(x, y))$. Here, the activation values and weight connections of neurons present in other processors are required to calculate the activation values of output neurons according to $O_k^P = \varphi(\sum_{i=1}^{M/P} \omega_{ki}^P \cdot H_i^P)$, with $k = 1, 2, ..., C$. In our implementation, broadcasting the weights and activation values is circumvented by calculating the partial sum of the activation values of the output neurons.
   b) *Parallel error back-propagation*. In this phase, each processor calculates the error terms for the local hidden neurons. To do so, *delta* terms for the output neurons are first calculated using $(\delta_k^o)^P = (O_k - d_k)^P \cdot \varphi'(\cdot)$, with $i = 1, 2, ..., C$. Then, error terms for the hidden layer are computed using $(\delta_i^h)^P = \sum_{k=1}^{P} (\omega_{ki}^P \cdot (\delta_k^o)^P) \cdot \varphi'(\cdot)$, with $i = 1, 2, ..., N$.
   c) *Parallel weight update*. In this phase, the weight connections between the input and hidden layers are updated by $\omega_{ij} = \omega_{ij} + \eta^P \cdot (\delta_i^h)^P \cdot f_j(x, y)$. Similarly, the weight connections between the hidden and output layers are updated using the expression: $\omega_{ki}^P = \omega_{ki}^P + \eta^P \cdot (\delta_k^o)^P \cdot H_i^P$.
3) *Classification*. For each pixel vector in the input data cube f, calculate (in parallel) $\sum_{j=1}^{P} O_k^j$, with $k = 1, 2, ..., C$. A classification label for each pixel can be obtained using the winner-take-all criterion commonly used in neural networks by finding the cumulative sum with maximum value, say $\sum_{j=1}^{P} O_{k^*}^j$, with $k^* = \arg\{\max_{1 \le k \le C} \sum_{j=1}^{P} O_k^j\}$.

IV. Experimental results

This section provides an assessment of the effectiveness of the morphological/neural classification algorithm described in Section II and its parallel implementation in Section III. First, we describe the hyperspectral image data set used in experiments. Then, we briefly describe the parallel computing architecture used for computational assessment. The setion concludes with a description of performance results for the proposed methodology, both from the viewpoint of classification accuracy and parallel efficiency.

A. Hyperspectral data

The image data set used in experiments was collected by the DAIS 7915 airbone imaging spectrometer of DLR. It was acquired at 1500 m flight altitude over the city of Pavia, Italy. The scene has a spatial resolution of 5 meters and total size of 400×400 pixels. Fig. 3(a) shows the image collected at 639 nm by the DAIS 7915 imaging spectrometer [13], which reveals a dense residential area on one side of the river, as well as open areas and meadows on the other side. Ground-truth is available for several areas of the scene (see Fig. 3(b)), comprising the following land-cover classes: (1) water; (2) trees; (3) asphalt; (4) parking lot; (5) bitumen; (6) brick roofs; (7) meadow; (8) bare soil; (9) shadows. Following a previous research study on this scene [6], we take into account only 40 spectral bands of reflective energy, and thus skip thermal infrared and middle infrared bands above 1958 mm because of low SNR in those bands.

B. Parallel computing architecture

The parallel computing architecture used in this work to illustrate performance of our parallel morphological/neural technique is a Beowulf cluster called Thunderhead and available at NASA's Goddard Space Flight Center in Maryland.

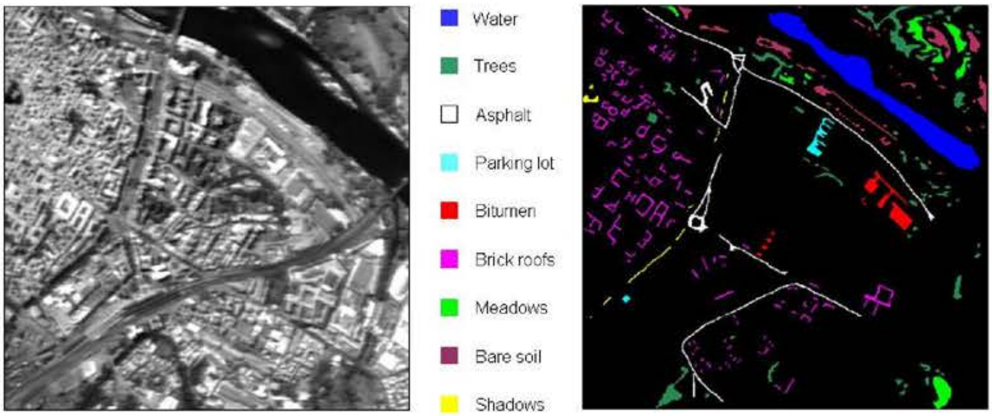

Fig. 3. (a) DAIS scene collected over the city of Pavia, Italy, and (b) Land-cover ground classes.

The system can be seen as an evolution of the HIVE (highly parallel virtual environment) project, started in 1997 to build a commodity cluster that was intended to be used by those who had not built it. The idea was to have workstations distributed among different locations and a large number of compute nodes (the compute core) concentrated in one area. The workstations would share the compute core as though it was apart of each. The HIVE was also the first commodity cluster to exceed a sustained 10 Gigaflop on a remote sensing algorithm. Presently, Thunderhead comprises 256 dual 2.4 GHz Intel Xeon nodes, each with 1 GB of memory and 80 GB of main memory. The total peak performance of the system is 2457.6 GFlops. Along with the 512-processor computer core, Thunderhead has several nodes attached to the core with 2 Ghz optical fibre Myrinet.

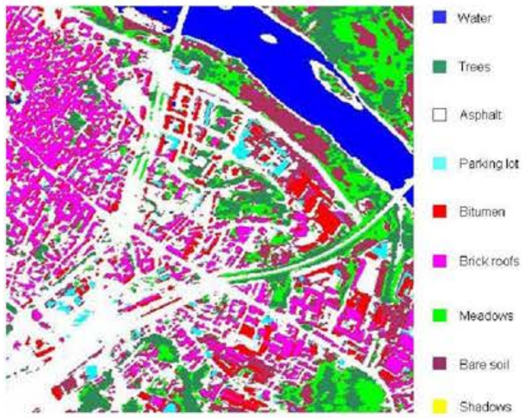

Fig. 4. Classification using the proposed morphological/neural algorithm.

C. Performance results

Before empirically investigating the performance of parallel hyperspectral imaging algorithms, we first test the classification accuracy of the proposed parallel morphological/neural classifier using the Pavia data set in Fig. 3(a). A random sample of 5% of the ground-truth pixels was first chosen from each of the 9 land-cover classes in Fig. 3(b). Morphological features were constructed for the selected training samples, and the resulting features were used to train a back-propagation MLP classifier with one hidden layer, where the number of hidden neurons was selected empirically. The number of hidden neurons was selected as the square root of the product of the number of input features and information classes. The trained classifier was then applied to the remaining 95% of labeled pixels in the scene, yielding the classification result depicted in Fig. 4.

For illustrative purposes, Table I shows the individual and overall classification accuracies obtained for each of the ground-truth classes. The table also includes the classification accuracies obtained using the full spectral information and PCT-reduced features as input to the MLP neural classifier. As shown by the table, morphological input features substantially improve individual and overall classification accuracies with regards to PCT-based features and the full spectral information. This is not surprising since morphological operations use both spatial and spectral information as opposed to the other methods which rely on spectral information alone. For illustrative purposes, Table I also includes (in the parentheses) the algorithm processing times in seconds for the different approaches tested, measured on a single processor in the Thunderhead system. Experiments were performed using the GNU-C/C++ compiler in its 4.0 version. As shown by Table I, the computational cost was slightly higher when morphological feature extraction was used.

To conclude this section, we investigate the properties of the parallel algorithm by timing the program on the Thunderhead Beowulf cluster, using 256 processor (the maximum number of processors available to us at the time of experiments). The measured execution times were in the order of 10 seconds for the morphological feature extraction part of the algorithm and about 50 seconds for the parallel MLP architecture. As a result, the proposed classifier was able to provide a highly accurate classification for the Pavia urban scene in about one minute. In this regard, the measured processing times represent a significant improvement over commonly used processing strategies for this kind of high-dimensional data sets, which

TABLE I

NUMBER OF TRAINING AND TEST SAMPLES AND CLASSIFICATION ACCURACIES (IN PERCENTAGE) ACHIEVED BY THE MORPHOLOGICAL/NEURAL CLASSIFIER USING MORPHOLOGICAL FEATURES, PCT-BASED FEATURES AND THE ORIGINAL SPECTRAL INFORMATION (PROCESSING TIMES IN A SINGLE THUNDERHEAD NODE ARE GIVEN IN THE PARENTHESES).

Class	**Training samples**	**Test samples**	**Spectral info (2981)**	**PCT-based features (3256)**	**Morphological features (3679)**
Water	114	4176	87.30	91.90	100
Trees	101	2444	94.64	93.21	98.72
Asphalt	85	1614	97,79	95.43	98.88
Parking lot	59	229	83.82	94.28	71.77
Bitumen	65	629	86.11	86.38	98.68
Brick roofs	106	2132	83.69	84.21	99.37
Meadow	62	1183	88.88	89.45	92.61
Bare soil	74	1401	79.85	88.24	95.11
Shadows	52	181	89.64	93.45	96.19
Overall accuracy	–	–	88.65	86.21	96.16

can take up to more than one hour of computation for the considered problem size, as indicated by the single-processor execution times reported on Table I.

Overall, experimental results in our study reveal that the proposed morphological/neural algorithm offers an accurate and scalable classification framework for texture-based analysis of urban hyperspectral images. Contrary to common perception that spatial/spectral feature extraction and back-propagation learning algorithms are too computationally demanding for practical use, results in this paper demonstrate that such approaches can provide accurate interpretation of complex, urban environments while, at the same time, being amenable for efficient parallel implementations, not only due to the regularity of the computations involved in both algorithms, but also because they can greatly benefit from the incorporation of redundant information to reduce sequential computations and involve minimal communication between the parallel tasks, namely, at the beginning and ending of such tasks.

V. CONCLUSION

In this paper, we have presented an innovative morphological/neural algorithm for texture-based classification of remotely sensed hyperspectral imagery collected over urban areas. Morphological operations are shown to be useful in order to preserve the relevant spatial/spectral information that allows for the separation of classes. Further, the proposed MLP neural architecture is shown to provide more accurate classification results when trained with morphological features instead of standard PCT-based features or the original spectral information in the data. Finally, we have also provided a parallel implementation of the proposed framework which has been specifically designed for massively parallel homogeneous platforms such as Beowulf clusters. Performance data in a parallel commodity cluster at NASA's Goddard Space Flight Center seem to indicate that the parallel algorithm is scalable and computationally efficient, although further work is still required to fully substantiate the above remarks.

ACKNOWLEDGMENT

The authors would like to thank J. Dorband, J. C. Tilton and J. A. Gualtieri for several helpful discussions regarding the development of parallel algorithms for the Thunderhead Beowulf cluster at NASA's Goddard Space Flight Center. The authors also gratefully acknowledge the imaging spectroscopy team at DLR for obtaining the DAIS 7915 image data in the framework of the EU HySens campaign.

REFERENCES

[1] P. Gamba, F. Dell'Acqua, A. Ferrari, J. A. Palmason, J. A. Benediktsson, J. Arnasson, "Exploiting spectral and spatial information in hyperspectral urban data with high resolution," *IEEE Geoscience and Remote Sensing Letters*, vol. 1, pp. 322–326, 2005.

[2] J. Chanussot, J. A. Benediktsson, M. Fauvel, "Classification of remote sensing images from urban areas using a fuzzy probabilistic model," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, pp. 40–44, 2006.

[3] D. A. Landgrebe, *Signal theory methods in multispectral remote sensing*, Hoboken: John Wiley and Sons, 2003.

[4] J. A. Benediktsson, M. Pesaresi, K. Arnason, "Classification and feature extraction for remote sensing images from urban areas based on morphological transformations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, pp. 1940–1949, 2003.

[5] J. A. Benediktsson, J. A. Palmason, J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, pp. 480–491, 2005.

[6] A. Plaza, P. Martinez, J. Plaza, R. Perez, "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, pp. 466–479, 2005.

[7] A. Plaza, D. Valencia, J. Plaza, P. Martinez, "Commodity cluster-based parallel processing of hyperspectral imagery," *Journal of Parallel and Distributed Computing*, vol. 66, pp. 345–358, 2006.

[8] P. Soille, *Morphological image analysis: Principles and applications*, Springer: Berlin, 2003.

[9] A. Plaza, P. Martinez, R. Perez, J. Plaza, "Spatial/spectral endmember extraction by multi-dimensional morphological operations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, pp. 2025–2041, 2002.

[10] J. Plaza, A. Plaza, R. Perez, P. Martinez, "Automated generation of semi-labeled training samples for nonlinear neural network-based abundance estimation in hyperspectral data," *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, pp. 345–350, 2005.

[11] C.-I. Chang, *Hyperspectral imaging: Techniques for spectral detection and classification*. Kluwer: New York, 2003.

[12] S. Suresh, S. N. Omkar, V. Mani, "Parallel implementation of back-propagation algorithm in networks of workstations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, pp. 24–34, 2005.

[13] A. Muller, P. Strobl, "HySens: DAIS/ROSIS imaging spectrometers at DLR," *Proceedings of SPIE*, vol. 5508, pp. 660–670, 2001.