Defence Science Journal, Vol 45, No. 4, October 1995, pp 285-291

Target Detection : Remote Sensing Techniques for Defence Applications

B.B. Chaudhuri and S.K. Parui

Indian Statistical Institute, Calcutta-700 035

ABSTRACT

The tremendous development in remote sensing technology in the recent past has opened up new challenges in defence applications. One important area of such applications is in target detection. This paper describes both classical and newly developed approaches to detect the targets by using remotely-sensed digital images. The classical approach includes statistical classification methods and image processing techniques. The new approach deals with a relatively new sensor technology, namely, synthetic aperture radar (SAR) systems and fast developing tools, like neural networks and multisource data integration for analysis and interpretation. With SAR images, it is possible to detect targets or features of a target that is otherwise not possible. Neural networks and multisource data integration tools also have a great potential in analysing and interpreting remote sensing data for target detection.

1. INTRODUCTION

Remote sensing image data of the earth's surface acquired from either aircraft or spacecraft platforms is widely available in digital format. Spatially, such an image comprises discrete picture elements, called pixels; and radiometrically, it is quantised into discrete brightness levels. Quite often several images of the same scene are captured in different spectral bands in a synchronous way. The spectral bands may range from visible to far-infrared spectrum. Sometimes images are taken in microwave range in synthetic aperture radar systems. The acquired images are initially processed for geometric and radiometric corrections.

Target detection is an important aspect in many defence applications. A target generally means an object or a region of interest and importance, such as urban areas, roads, railway tracks, shipyards, etc. The task in target detection is to locate and label a particular target among non-targets in a scene. This task has been extensively studied in computer vision where the main steps are classification of the scene into regions of interest, looking for potential structures, making geometric analysis of these structures and finally confirming the presence or absence of a target using proper knowledge base. But, its application in remote sensing had been limited because of the inadequate ground resolution of the image available in the past. However, with resolution better than 30-40 m provided by the current remote sensing technology, it is now possible to a great extent to identify even smaller targets, like bridges, runways and buildings. Consequently, the scope of application of remotely-sensed imagery in defence has considerably widened.

Most of the pattern recognition and interpretation techniques developed for remote sensing require multispectral image data. Here the classification of individual pixels is the primary task to which several approaches exist. Of these two main approaches are described in Sec. 2. It is also possible to detect targets in remotely-sensed images with a single spectral band for which a host of image processing and analysis techniques are available. Some of these techniques are described in Sec. 3. Multisource data integration and SAR imaging approaches are briefly discussed in Secs 4 and 5, respectively.

Received 08 March 1995

2. SPECTRAL CLASSIFICATION APPROACH

One of the most widely used applications of pattern recognition techniques to image classification has been the assignment of individual pixels to land cover categories or information classes on the basis of their multispectral values[1,2]. Here a pixel is looked upon as a point in the multidimensional space. The groups of pixels in this space which are homogeneous in some sense are called spectral classes. Remote sensing is successful because in many instances the spectral classes coincide with the information classes.

A variety of approaches is available for spectral classification, ranging from those using probability distribution models for the classes of interest to those in which the multispectral space is partitioned into class-specific regions based on optimally located surfaces. The essential practical steps that are common to these different approaches are:

(a) Deciding the set of land cover categories or information classes into which the image is to be classified or segmented. These classes could, for example be water, croplands, asphalt areas, concrete structures, etc,

(b) Selecting a group of representative or prototype pixels from each of these information classes. These groups of pixels form the training set,

(c) Using the training set of pixels to estimate the parameters of the particular classifier algorithm to be used. These parameters could be the properties of a probability distribution model or could define the equations determining surfaces in the multispectral space or could be the weights in a neural networks model, and

(d) Using the trained classifier, classifying every pixel in the image into one of the desired information classes.

The main step in designing a classifier for pixel classification is the task of training mentioned in (c) above. Two approaches to multispectral classification of pixels i.e., the conventional statistical approach and the neural networks approach are presented here.

2.1 Statistical Approach to Classification

Suppose the spectral classes are C_i, $i = 1,2, \ldots, M$ and the corresponding probability density functions are $p(x{:}C_i)$ where the vector x indicates the spectral vector. There are several statistical techniques for classifying a spectral vector into one of the M spectral classes. Two most widely used techniques are:

2.1.1 *Maximum Likelihood Classification*

Given an arbitrary vector x, it is necessary to compute the conditional probabilities $p(C_i{:}x)$ for $i = 1,2, \ldots, M$. The probability $p(C_i{:}x)$ gives the likelihood that the pixel with spectral vector x actually comes from C_i. The classification rule in this case is to classify x into C_i if $p(C_i{:}x) > p(C_j{:}x)$. $\forall_j \# i$ This simple classification rule is a special case of a more general rule, namely Baye's classification in which the decisions can be biased according to different degrees of significance being associated with different incorrect classifications.

Now the quantities $p(C_i{:}x)$ are unknown. To make an estimate of these quantities, one should have, for each class C_i, a training set of pixels with their spectral vectors x. This can be used to estimate the probability density function $p(x{:}C_i)$ for each class. The form of this function is normally known a priori and only its parameters are unknown. The maximum likelihood estimates of these parameters are those values of the parameters that maximise the joint probability density function $\pi_k p(x_k{:}C_i)$, where $|x_k|$ are the spectral vectors from the training set representing the ith class.

Now the estimated $p(x{:}C_i)$ and the desired $p(C_i{:}x)$ are related by Baye's theorem as $P(C_i{:}x) = p(x{:}C_i)$ $p(C_i)/p(x)$ where $p(C_i)$ is the probability that the ith class occurs in the image and $p(x) = \Sigma_i\, p(x{:}C_i)p(C_i)$. The quantities $p(C_i)$ and $p(C_i{:}x)$ are known as prior and posterior probabilities, respectively. It is commonly assumed that the functions $p(x{:}C_i)$ have the form of multivariate normal models.

2.1.2 *Minimum Distance Classification*

The effectiveness of the maximum likelihood classifier depends on a proper choice of the form of $p(x{:}C_i)$ and on reasonably accurate estimation of its parameters. When the form of $p(x{:}C_i)$ is unknown or when the number of training pixels is too few to estimate all the parameters of $p(x{:}C_i)$, it may be advisable to design a classifier that is based only on the mean spectral vector of a class. The minimum distance classifier is one such tool where the mean spectral vector of each class is estimated from the training set and a new pixel is classified in the class with the mean vector that is closest to the spectral vector of the pixel.

2.2 Neural Networks Approach to Classification

In the last few years, there has been a surge of interest in artificial neural network models with the aim of achieving human-like performance in decision making, particularly in the context of speech and visual patterns[3]. These models are composed of many non-linear computational elements operating in parallel. The computational elements or nodes are connected via weights that are adapted during use to improve performance. A neural network model is specified by its net topology, node characteristics and training or learning rules. Instead of executing a program of instructions sequentially as in von Neumann computers, these models respond, in parallel, to the inputs that are presented to them. The result is not stored in any specific memory location, but consists of the overall state of a network after it has reached some equilibrium situation. Artificial neural network models can be used for pattern classification[4]. Most commonly used models for classification are single layer and multilayer perceptrons[5].

When the classes under consideration are linearly separable, a single layer perceptron is enough for classifying an arbitrary feature vector into one of the two classes[6]. But when the classes are not linearly separable, a multilayer perceptron is normally used for classification since such a network with appropriate weights can form arbitrarily complex decision regions in the feature space.

2.2.1 Single Layer Perceptrons

Suppose the classification problem under consideration is in a p-dimensional feature space. For a two-class problem ($M = 2$), a single layer perceptron specifies a (p–1) = dimensional hyperplane which discriminates between the two classes in an optimal way. Training a single layer perceptron means getting hold of the parameters of this hyperplane on the basis of a training set. These feature vectors are sequentially presented to the network and they modify the connection weight vector w by shifting it in the direction of the input feature vector[7]. This process of modifying the weight vector is continued until it stabilises. The original perceptron convergence procedure[8] updates the weight vector by a fixed incremental change in the following way:

The feature vector at time t is denoted by $x(t)$. Suppose the true class identification associated with $x(t)$ is $d(t)$ whose value is 0, if $x(t)$ is from class 1, and 1 if $x(t)$ is from class 2. The weight vector is a p-dimensional vector and is, at time t, denoted by $w(t)$. The output at time t, denoted by $y(t)$, is defined as $y(t) = h(f(w(t), x(t))$, where h is the hard limiting activation function so that $h(f)$ is either 0 or 1 depending on whether $f<0$ or $f\geqslant 0$ and $f(w(t),x(t)) = (w(t))^T x(t)$, where T indicates transpose. At time t, the weight vector $w(t)$ is updated to $w(t+1)$ as $w(t+1) = w(t) + \eta\,[d(t) - y(t)]x(t)$, where η is a gain term lying between 0 and 1. The initial weights are small random values. To achieve convergence of $w(t)$, η should decrease with time t and should satisfy certain conditions[6]. But an important problem encountered in practice is about the rate of convergence of $w(t)$[9].

2.2.2 Multilayer Perceptrons

For more complex problems, where single layer perceptrons fail, multilayer perceptrons are more appropriate for classification. They have one or more layers of hidden nodes and one or more outputs nodes. The commonly used algorithm to train a multilayer perceptron with sigmoidal non-linearity is backpropagation[10]. Such a net after being trained forms the required decision regions bounded by smooth curves. During training, the system first uses the input vector to produce its own output vector and compares this with the desired or target output vector (of dimension M). If there is any difference, modifications of the connection weights are made on the basis of what is called the generalised delta rule. This rule calculates an error function as follows:

$$E = \frac{1}{2N} \sum_{r=1}^{N} \sum_{k=1}^{M} (t_{rk} - o_{rk})^2$$

where t_{rk} and o_{rk} are respectively the kth components of the target and calculated output vectors for the input pattern r and N is the size of the training set. It changes the weights in a manner to reduce the error E as quickly as possible. The convergence of training procedure is achieved by considering the incremental change in individual weight components ω_i as $\Delta\omega_i = -\eta\delta E/\delta\omega_i$ where η is a gain term which controls the rate of learning However, for practical problems, this training process may be extremely slow. The choice of optimal η to achieve fast convergence of the backpropagation training algorithm is an open problem. Active research is going on in this direction[11].

One possible application of multilayer perceptrons is in detecting structures in remotely-sensed imagery[12]. Figure 1 shows a 256 × 256 window of an infrared-band image from IRS-1A satellite. The line structures in the image detected by a multilayer perceptron are shown in Fig. 2. Here the training set consists of 100 pixels lying on horizontal and vertical road segments and 100 non-road pixels and their neighbourhoods.

Figure 1. An infrared-band image from IRS-1A satellite.

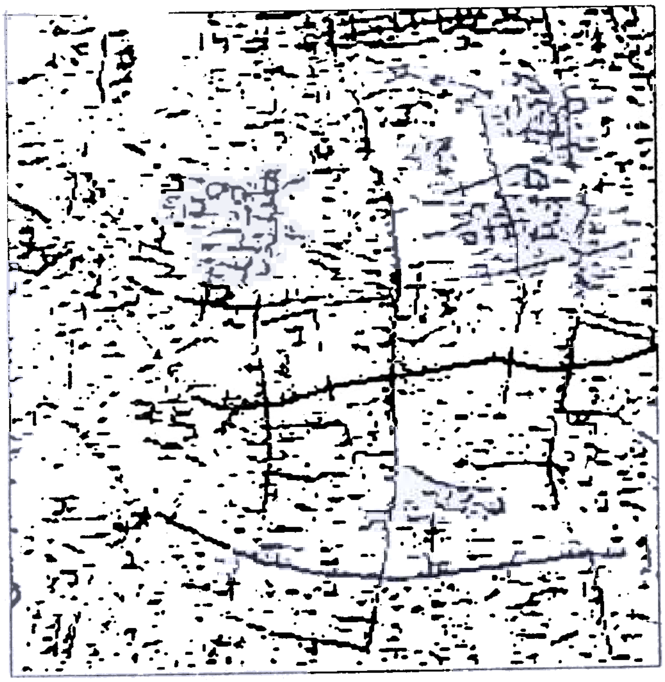

Figure 2. Lines detected in the image in Fig. 1 by a multilayer perceptron.

3. GEOMETRIC APPROACH

In many situations, the geometric features of objects of interest play an important role in their detection. To identify these features in a gray level image, the image is first enhanced and then segmented. The enhancement techniques are characterised by operations over neighbourhoods around individual pixels. If a geometric feature has an area then its edges are enhanced. On the other hand, if the feature is a line structure, the lines are enhanced. The enhanced output is again a gray level image.

3.1 Edge Detection

Given below are four spatial masks for enhancing edges or meaningful discontinuities in gray level in four possible orientations:

−1 0 +1	−1 −1 −1	0 +1 +1	+1 +1 0
−1 0 +1	0 0 0	−1 0 +1	+1 0 −1
−1 0 +1	+1 +1 +1	−1 −1 0	0 −1 −1
vertical	horizontal	diagonal	

The computed gray level value is associated with the central pixel of a mask. Thus, for every pixel (except the boundary pixels) in the image, there are four scores from the four masks. The maximum among these scores is the enhanced gray level of the pixel.

Another way to enhance the edges is by computing a local derivative operator of gray levels. One such set of gradient operators, known as Sobel operators, is defined by two masks shown below.

−1 2 1	−1 0 1
0 0 0	−2 0 2
−1 −2 −1	−1 0 1
horizontal	vertical

3.2 Line Detection

Four masks normally used to enhance (1-pixel thick) lines in a gray level image are:

−1 2 −1	−1 −1 −1	2 −1 −1	−1 −1 2
−1 2 −1	2 2 2	−1 2 −1	−1 2 −1
−1 2 −1	−1 −1 −1	−1 −1 2	2 −1 −1
vertical	horizontal	diagonal	

The scores from these masks are additive and pose some problems in line detection[13]. A multiplicative score proposed instead to overcome these problems will now be explained. For a horizontal mask shown below the enhanced gray level output g for the central pixel is:

$$\begin{matrix} b_1 & b_2 & b_3 \\ a_1 & a_2 & a_3 \\ c_1 & c_2 & c_3 \end{matrix} \quad + \sqrt{(A-B)(A-C)}$$

if $(A-B)(A-C)>0$ and $-\sqrt{-(A-B)(A-C)}$ otherwise, where $A = (a_1+a_2+a_3)/3$, $B = (b_1+b_2+b_3)/3$ and $C = (c_1+c_2+c_3)/3$ The image g is defined for each direction in which lines are to be detected. In each such direction, g gives a directional differential image in which dark lines in a bright background or bright lines in a dark background will show positive g values with reasonably high magnitude. For pixels in the areas of nearly uniform original gray values or around the boundary of thick objects, g values will be close to zero. For pixels in the areas of monotonically increasing or monotonically decreasing (in the direction perpendicular to the direction of the mask) gray values, g will have negative values.

The g images in several directions can be properly segmented to detect linear structures, like roads, runways, bridges that may be present in a remotely-sensed image[14]. For example, the roads detected in the IRS-1A image in Fig. 1 using this method are shown in Fig. 3.

Figure 3. Lines detected in the image in Fig. 1 on the basis of the multiplicative score, g.

4. MULTISOURCE DATA INTEGRATION APPROACH

In spectral classification, the pixels in an image are classified independently of the classifications of their spatial neighbours. Techniques are available for classifying pixels in the context of their neighbours[15]. These require information from a spatial model of the image and tend to produce a better classification since this is consistent both spectrally and spatially.

This integration of spectral and spatial information can be extended further. One such extension is multisource data integration in remote sensing which will be a real challenge in the near future[16]. New instruments and new sensors give rise to a large variety of new views of the real world. This huge amount of data has to be combined and integrated into a model of this world. Also, to meaningfully interpret these data, one has to have information about how the data are collected and what their characteristic properties are. Multiple sources provide complementary views of the world. Integration of information from these views throws up new possibilities in target detection.

With the recent advances in sensor technology, the number of different sensor platforms that carry imaging payloads has increased tremendously. These sensors produce data covering different portions of a broad range of the electromagnetic spectrum at different spectral and spatial resolutions, providing the users with enormous amount of useful information. These data are heterogeneous in their format, radiometric characteristics, geometric properties and temporal sampling. To fully exploit these increasingly sophisticated multisource data, advanced data fusion techniques become essential.

Data fusion techniques can be of three types depending on the stage at which fusion takes place. These types are pixel-level, feature-level and decision-level fusion techniques. Pixel-level fusion techniques generate new pixels with a pre-selected spatial resolution common to all data sources involved. Image registration is a typical example of pixel-level data fusion. In feature-level data fusion, generally image analysis techniques are first employed to extract some features from each data source independently. Then the data integration is done on the basis of these features[17,18]. For example, edge or line structures can be simultaneously segmented from multiple images and can then be integrated to detect targets more reliably. Decision-level fusion means integration of several interpretations obtained from different data sources to arrive at a consensus interpretation.

5. SAR IMAGING APPROACH

Visible band imaging systems act poorly if there is a cloud cover or lack of illumination over the scene. The synthetic aperture radar (SAR) imaging system can overcome this problem by illuminating the scene by microwave, typically in *X*-band which can penetrate the cloud. The signal scattered by the scene is received by the sensor and the variation of the received signal strength generates the SAR image of the scene which can be used to detect a variety of targets[19]. A typical SAR image contains speckle noise that makes the granular appearance over the whole image. Considerable effort is required to suppress the speckle noise either before or after the image formation[20]. Speckle suppression before the image formation is done mainly by multilook processing while various spatial domain and frequency domain filtering techniques are applied on the already received image. However, the processed image still contains noise and the visual quality of SAR and visible band images are markedly different because of speckle texture. For further processing say for edge detection and region segmentation, texture analysis-based techniques are quite promising.

Apart from visibility under cloud and at night, the SAR system has an advantage of detecting objects in motion by the Doppler shift method. Thus, a moving train will remain off the railway track (but parallel to the track) to an extent that depends on its speed. The SAR signal is strong enough to extract objects with sharp bend even if the objects are poorly visible. Thus, the SAR image can complement the visible band image and integration of information from both types of image would be most effective in target identification.

REFERENCES

1 Richards, J.A. Remote sensing digital image analysis: An introduction, Springer-Verlag, Berlin, 1986.

2. Parui, S.K.; Shankar, B.U.; Datta, A. & Majumder, D.D. Unsupervised classification of Indian remote-sensing satellite imagery, Proceedings of the 3rd. International Conference on Advances in Pattern Recognition and Digital Techniques, Calcutta, 1993, pp. 68-74.

3. Carpenter, G.A. & Grossberg, S. (Eds) Neural networks for vision and image processing, MIT Press, 1992.

4. Pao, Y.H. Adaptive pattern recognition and neural networks, Addison-Wesley, 1989.

5. Minsky, M. & Papert, S. Perceptrons, MIT Press, 1969.

6. Duda, R.O. & Hart, P.E. Pattern classification and scene analysis, Wiley, New York, 1973.

7. Lippmann, R.P. An introduction to computing with neural nets. *IEEE ASSP Magazine*, 1987, pp. 4-22.

8. Rosenblatt, R. Principles of neurodynamics, Spartan Books, New York, 1959.

9. Bhattacharya, U. & Parui, S.K. On the rate of convergence of perceptron learning, Pattern Recognition Letters, 1995, **16**, pp. 491-97.

10. Rumelhart, D.E.; Hinton, G.E. & Williams,R.J. Learning internal representations by error propagation. In parallel distributed processing: Explorations in the microstructure of cognition, D.E. Rumelhart & J.L. McClelland (Eds), MA, MIT Press, Cambridge, 1986.

11 Bhattacharya, U. & Parui, S.K. Self-adaptive learning rates to speed up convergence of backpropagation algorithm. KBCS Technical Report, Indian Statistical Institute, 1994.

12. Mukherjee, A.; Bhattacharya, U.; Parui, S.K. & Chaudhuri, B.B. A neural networks approach to detect the linear features in remotely-sensed images. KBCS Technical Report, Indian Statistical Institute, 1994.

13. Parui, S.K.; Shankar, B.U.; Mukherjee, A. & Majumder, D.D. A parallel algorithm for detection of linear structures in satellite images. Pattern Recognition Letters, 1991, **12**, pp. 765-70.

14. Mukherjee, A.; Parui, S.K.; Chaudhuri, B.B. Krishnan, R. & Rao, K.K. Detection of linear features in satellite imagery using robust estimation. Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, 1994 pp. 514-16.

15. Swain, P.H.; Varderman, S.B. & Tilton, J.C. Contextual classification of multispectral image data. Pattern Recognition, 1981, **13**, pp. 429-46.

16. Tilton, J.C. (Ed). Proceedings of the workshop on Multisource Data Integration in Remote-sensing, sponsored by IEEE and NASA, NASA Conference Publication 3099, 1991.

17. Shufelt, J. & McKeown, D.M. Use of information fusion to improve the detection of man-made structures in aerial imagery. Proceedings of the workshop on Multisource Data Integration in Remote Sensing, J.C. Tilton (Ed), NASA Conference Publication 3099, 1991, pp. 94-110.

18. Kropatsch, W.G. Integration of SAR and DEM data — geometrical considerations. Proceedings of the workshop on Multisource Data Integration in Remote Sensing, J.C. Tilton (Ed), NASA Conference Publication 3099, 1991, pp. 27-38.

19. Derin, H. *et al.* Modelling and segmentation of speckled images using complex data, *IEEE Trans. Geosci. Rem. Sensing*, 1990, **GE-28**(1), pp. 76-87.

20. Crimmins, T. Geometric filter for speckle reduction. *Appl. Optics*, 1985, **24**, pp. 1438-43.