Applying Artificial Neural Networks to Generate Radar Simulation Data Bases

Harry H. Heaton III
Science Applications International Corporation
Dayton, Ohio

ABSTRACT

Modern combat aircraft sensor systems such as synthetic aperture radar (SAR) produce highly detailed, information rich displays. The simulation of such displays for training has demanded ever increasing computational resources as well as data sources more detailed than normally available digital feature analysis data (DFAD). By focusing on the correct reproduction of the content of a radar display rather than on a detailed model of radar physics, a novel Digital Radar Land Mass Simulator (DRLMS) for training is briefly described. A prototype of the system reproduces realistic real-beam, Doppler beam sharpened (DBS), and SAR ground maps from readily available data sources.

This radar simulation technique depends upon highly detailed, modified phototexture databases which contain both dimensional and effective radar cross-section information for broad area clutter and specific radar targets. This paper discusses the application of artificial neural networks in generating such databases from readily available data sources including Project 2851 and commercial satellite data. The issues, differences and solution approaches necessary to generate databases from such disparate sources as overhead imagery, DFAD feature data and existing simulator visual system databases are examined.

The techniques discussed have broad applications to the low-cost simulation of imaging sensor displays including millimeter microwave (MMW) and forward looking infrared (FLIR). The approach also drastically reduces the computational needs for a DRLMS system. The prototype, capable of generating SAR maps, was hosted on a single Motorola 68040 processor in a Macintosh personal computer. A simulation of the APG-68 radar, including real beam, expanded and DBS modes, is targeted to run in real time on a single MIPS R-4400 microprocessor.

ABOUT THE AUTHOR

Mr. Heaton has been an employee of Science Applications International Corporation (SAIC) for the past ten years, where he is the manager of the Microprocessor Processor Applications Branch at SAIC's Aeronautical Systems Operation in Dayton, Ohio. Mr. Heaton has been involved in the development of several simulation systems, including the B-1B Engineering Research Simulator, the B-1B High Fidelity Defensive Crewstation Simulator, and the Ground Support System for the F-16 On-Board Electronic Warfare Simulator. He is currently the program manager for an effort to field a low-cost ground map radar simulator for the APG-68 radar used by the F-16 aircraft.

INTRODUCTION

The Digital Radar Land Mass Simulator (DRLMS) discussed below departs from the traditional approach used for most DRLMS. Rather than relying on a rigorous simulation of the electromagnetic properties of the radar, the approach below uses photographic imagery pre-processed using a neural network to yield a database of pre-assigned radar reflectivities, which are then manipulated in real-time to yield a simulated high resolution Synthetic Aperture Radar (SAR) patch. For the engineer familiar with the physics of radar systems, this approach may seem too simplistic a process to yield credible results. Yet the images produced, while not to be confused with a target signature prediction, are convincing for man-in-the-loop training and laboratory applications.

One factor that allows the simplified DRLMS approach is the nature of image formation in high-resolution radar, where the effects of aspect are reduced in comparison to low resolution systems. This reduction in aspect sensitivity allows effective simulated imagery to be generated from simplified processes.

HIGH RESOLUTION RADAR TARGETS

For ground mapping radar, nearly all point targets extend across several wavelengths, and are said to occupy the optical region, where the ray tracing methods of geometric optics can be applied to estimate the radar cross section (RCS). A radar target is composed of one or more scatterers, depending upon the nature of the target and the radar. For the purpose of RCS estimation, these individual scatterers are described using various geometric shapes that allow ray tracing. Several of the fundamental shapes and their effective RCS (σ) in the optical region are defined below:

Sphere of radius a $\quad \sigma = \pi a^2$

Normal to flat plate of areaA $\quad \sigma = \frac{4\pi A^2}{\lambda^2}$

Normal to triangular corner

reflector of edge length a $\quad \sigma = \frac{4\pi a^4}{3\lambda^2}$

The spherical scatterer is considered to be isotropic, and thus presents the same radar cross section regardless of the incidence angle of illumination. For flat plate reflectors and complex reflectors made up of flat plates, such as a corner reflector, the return strength varies depending upon the incident angle. The RCS of a flat plate exhibits a sensitivity to aspect (directivity) that is proportional to the size of the reflector. Table 1 depicts the mainlobe null-to-null angle for flat plate reflectors of increasing size. Note that for small reflectors, the useful angle is quite large.

Table 1:
Flat Plate Directivity vs/ Size (X-Band)

Flat Plate Size (m)	Flat Plate Size (λ)	Null-to-Null (degrees)
0.1	3	$\approx 38^*$
.3	10	≈ 12
3	100	≈ 1
30	1000	≈ 0.1

* value shown does not consider resonant effects applicable to targets of less than 10λ.

Complex reflectors such as trihedral corner reflectors are even less sensitive to aspect. Large corner reflectors (such as found in many man-made objects) presenting strong returns over incident angles of roughly 45¡.

For a high-resolution X-band radar with a resolution cell size of 1 meter, each cell can only contain a few optical region scatterers (the cell is only $\approx 30\lambda$ on a side). Since the most probable scatterer size in a cell is small in terms of λ, the mainlobe size is correspondingly large, resulting in a return that is relatively constant over large aspect angles. This is the case found in practice with high resolution radar, where scattering sources are found to remain fixed in target location over aspect variations up to about 30¡ [1]. The small size of the cell in λ also increases the probability that a single scatterer dominates the return.

In contrast, a low resolution radar, where each resolution cell covers a very large area relative to λ, contains multiple scatterers in the optical region. First, it is now possible for very large scattering facets to exist within the range cell, resulting in strong, but highly directive returns.

Also, the return from multiple reflectors within the range cell is the phasor sum of the individual returns. This sum varies depending upon the geometry between the radar antenna and the relative phase between the individual scatterers. This phasor sum can vary greatly as the aircraft moves relative to the target, resulting in fluctuating return intensities that are highly aspect sensitive. This case is illustrated in Figure 1, where an identical set of corner reflectors is illuminated by both a high resolution and a low resolution radar. While the low resolution radar return is dependent upon the phasor sum, the high resolution radar sees relatively constant returns from fixed locations over large aspect angles.

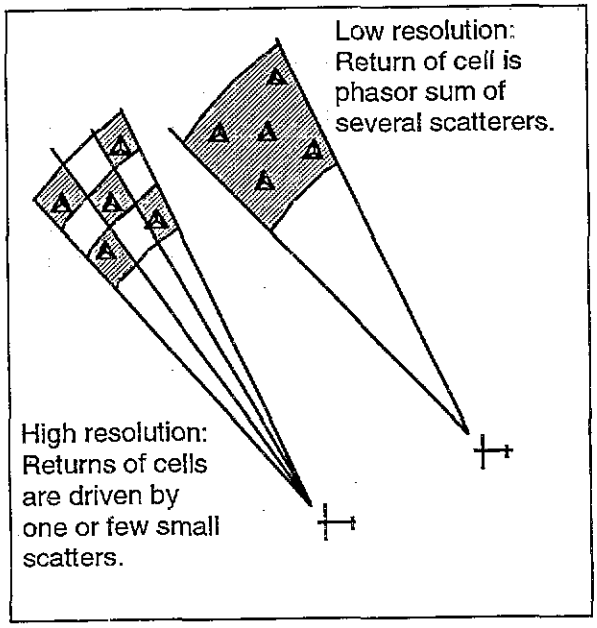

Figure 1: Comparison of return from high and low resolution radar from discrete corner reflectors.

The astute reader may wonder by this point, "What on Earth has this got to do with neural networks and DRLMS data bases?" The point of the preceding discussion is to identify the underlying reasons for the fundamental differences in the displayed returns from real-world targets produced by both high and low resolution radar systems. By capitalizing on the fact that the spatial arrangement of scatterers that are relatively insensitive to aspect (compared to low resolution radar) it is possible to use readily available photographic image sources to construct data bases for effective high resolution radar simulation. The photographic nature of the data inherently provides the spatial arrangement of features.

Neural networks can be applied to identify material classes from the spectral signature of readily available multi-spectral imagery. This can in turn, now be assigned a suitable value to represent radar reflectivity. In our simplified DRLMS process, we have applied this to imagery on a pixel-by-pixel basis, allowing a detailed correlation between a simulated SAR patch and the original photographic data source.[2] Below, the use of this approach is described and contrasted to traditional DRLMS processing.

TRADITIONAL DRLMS APPROACH

Conventional DLRMS systems have relied upon the availability of digital representations of terrain and cultural features. The terrain is maintained as DTED terrain posts and cultural data is derived from DFAD text files. A propagation model determines the two-way losses for each pulse as it travels to and from terrain and features, calculating the phasor sum for the signal at the receiving antenna. The energy reflected from the terrain is determined by ray-tracing each pulse, taking into accounting for such attributes as slope, material type, and terrain texture and correcting for receiver attributes such as sensitivity time control (STC) and gain. Because of the complexity of these calculations, DRLMS systems have been synonymous with complex, special purpose hardware in order to maintain real-world display update rates. Extending this approach to high resolution radar such as SAR continues to challenge the state-of-art in computing.

A key limitation for high resolution DRLMS is the content of the underlying database[3]. Advanced signal processing technologies allow radar such as Doppler beam sharpened (DBS) and synthetic aperture radar (SAR), to achieve resolutions on the order of a 5 meters or less for tactical systems. Standard database products have not kept pace. Level 2 DTED for example, provides terrain at 30 meter resolution, and Level 2 DFAD allows cultural resolution to two meters. While suitable for some high resolution

radar simulations, the Level 2 DFAD product has limited geographic availability. Military training simulators frequently have a world-wide mission requirement.

To fill the gap between the expected content of the simulated sensor display and the content of the typical database, generic patterns are often applied. The major drawback of this approach is the loss of geo-specific content in the data base. Unless modified by specific cultural features, a SAR patch of one urban area will look much the same as the next, and may bear little resemblance to the real-world.

PHOTOTEXTURE BASED DRLMS

Phototexture is the term applied here to multi-spectral aerial images. (The term "phototexture" comes from visual simulation where it describes photographic data that is applied to polygon surfaces to enhance realism.) Such imagery is readily available, and is included as part of the Project 2851 SIF format. This same phototexture data can be applied as the starting point in generating representative high resolution radar imagery. Typically, these images are obtained from commercial satellite or aerial photography.

Prototype system description

SAIC has developed a prototype DRLMS system that demonstrates the utility of this approach to high-resolution radar simulation. The prototype radar simulation system is hosted on a commercial Macintosh Quadra computer (68040 @ 25 MHz). No special purpose hardware is required, allowing for re-hosting on a variety of systems. Pre-processing of the raw phototexture data followed by selective real-time image processing is the core of the system. The overall process for generating a SAR image is depicted in block diagram form in Figure 2. With additional processing to simulate fluctuating target returns and other effects, the capability has been extended to DBS and real-beam displays as well.

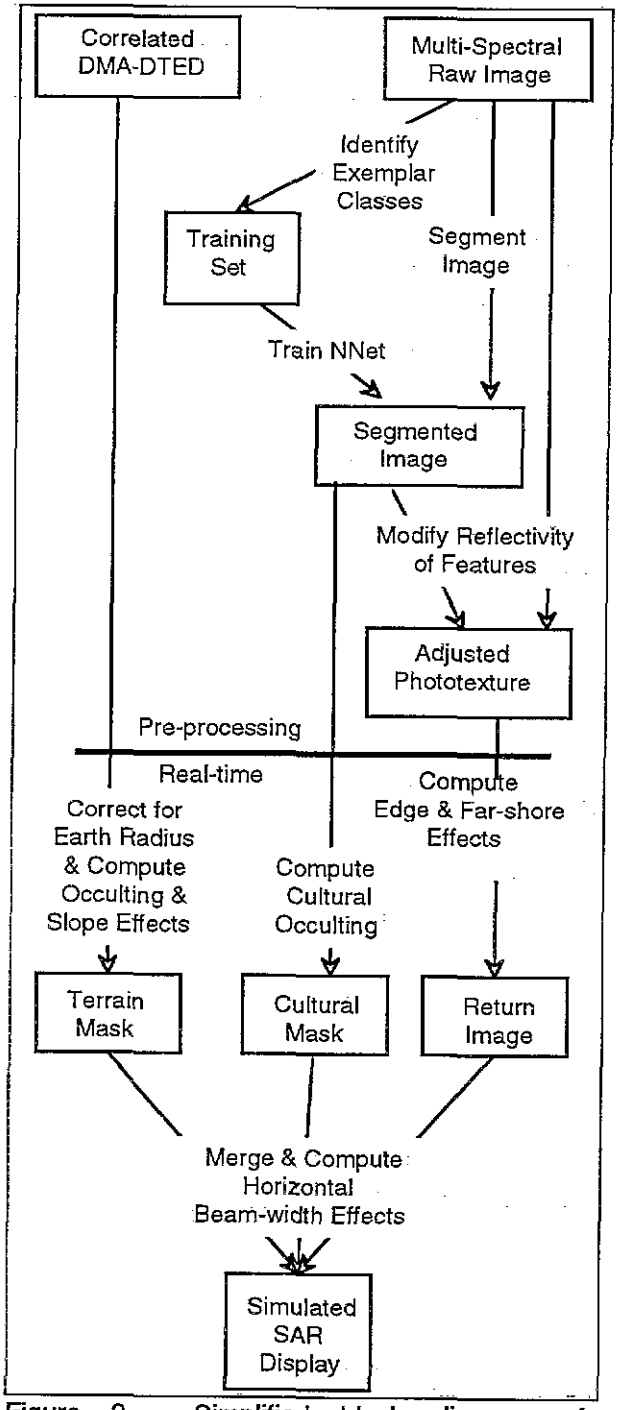

Figure 2. Simplified block diagram of phototexture-based radar simulation process. Processes depicted above the heavy line are pre-processing. Operations below are performed in real-time.

Database preparation overview

In comparison to Level 2 DFAD, high resolution photographic imagery is available on virtually a world-wide basis. Remote sensing satellites such as SPOT routinely generate multi-spectral imagery with a resolution of 20 meters, and panchromatic imagery to 10 meters. Imagery sold commercially from Russian satellites with the MK-4 camera provide multi-spectral six meter data. Recent easing of U.S. restrictions on the resolution of commercial satellite imagery foreshadows a blossoming market for sub-10 meter imagery. In addition to satellite imagery, false color and multi-spectral imagery from aircraft offers affordable access to sub-meter photographic data sources.

The DRLMS simulation process begins with the segmentation of a multi-spectral photographic image into feature classes. Segmentation is the process of breaking a complex data set, such as a multi-spectral image, into distinct sub-classes. This is accomplished by training a feed-forward, back-propagation neural network to classify individual pixels based upon their spectral content in the available bands[2,4]. This type of network is an iterative gradient algorithm that is trained to minimize the mean-square error between its output and the desired result, as characterized by a set of exemplar data. Figure 3 depicts the general configuration of such a network.

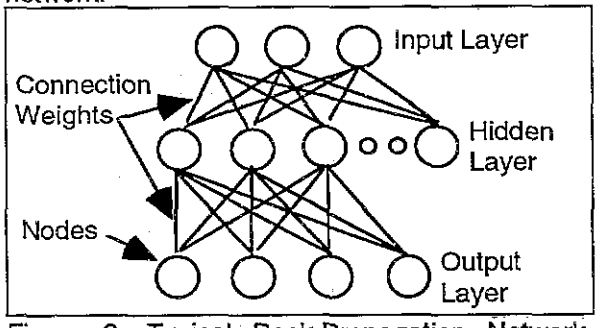

Figure 3: Typical Back-Propagation Network Configuration.

The number of input nodes corresponds to the number of spectral bands-per-pixel while the number of output nodes corresponds to the desired number of feature classes to be identified. The hidden layer allows the network to develop internal representations of the input data for re-mapping into the output classes. The number of hidden layer nodes is less deterministic, and depends greatly upon the nature of the input image data as discussed below. In Figure 3, each circle represent a processing node, with the connecting lines depicting the interconnection weights. Each node simply outputs a value which is dependent upon the sum of inputs. The connection weights (which are initially random) are adjusted during training until the output nodes generate a desired value in response to a known input.

To segment an image, the neural network is first trained on an exemplar set of pixels identified in the image. The exemplar set is identified manually by a photo-analyst, and includes pixels coded for a variety of different reflectivity features. Examples include bare soil, asphalt surfaces, cultural structures and water. The network is trained to a suitably low error value, then used to segment the entire image into the identified classes on a pixel-by pixel basis. To aid in the visualization of the segmented data, each feature class is assigned a pre-determined pixel value. The resulting image is inspected for systematic errors in classification, which are corrected manually. Minor errors in classification (such as the coding of a high-reflectivity pixel in the midst of an open field) are not removed. These artifacts become part of the background clutter that is present in a real-world scene.

In practice, we have used the segmented imagery to develop phototexture for radar processes in two different ways. Both depend upon the pre-assignment of values to pixels that represent displayed intensities of radar returns expected from typical grazing angles.

In our first approach, the segmented phototexture was used only to remove gross artifacts and adjust the underlying pixel values of the original phototexture. For example, water pixels would be set to a uniform low value (representing the return expected from a large forward scattering surface) while metallic structures are brightened. More recently, we have used the segmented image to define the geometric limits for Gaussian noise patterns with distributions appropriate for scatterers that display a Rayleigh distribution. The former method has the advantage of yielding a more detailed scene content, at the expense of

inappropriate returns from some features. The latter method provides a more predictable, but less detailed return. For terrain processing, correlated DTED is used to provide underlying land form data.

Neural network application

Several different neural networks paradigms can be applied to process multi-spectral images to identify, on a pixel-by-pixel basis, the probable material class of the surface to support the processes outlined above. We have selected supervised networks such as the feed-forward back-propagation network described above over unsupervised clustering paradigms, although both have been applied to image segmentation[2]. Similar techniques are used in remote sensing to identify ore bodies and estimate crop yields from multi-spectral data. For the sample images here, a three layer back-propagation neural network was applied. The essential steps in the process are:

- extracting a suitable training set from the image
- configuring and training the neural network
- processing the raw image
- manual correction of classification errors

The sample image in Figure 4 is typical in that it contains a mixture of man-made structures and natural features. Through the segmentation process, each pixel is assigned to a pre-defined feature class. In the example here, the image was segmented into six classes, including soil, asphalt, concrete, man-made structures, vegetation, and water. The exemplar pixel groups used to construct the training set were identified through manual image analysis, with care taken to represent the variation present in the original image.

It is a frequent misconception that the exemplar set should consist of "perfect" examples of each class. Actually, the reverse is true. For example, the metallic structure exemplar pixel set includes both pixels for large industrial structures as well as for smaller buildings as the intent during processing was to assign both structures to a single class. The set of exemplar pixels for asphalt includes both pixels from taxiways as well as for the small secondary roads. In training a neural network, it is important to attempt to include exemplar pixels for the entire range of values expected for a particular class. This requirement is reduced in practice by corrupting the exemplar set data with random noise while the network is training. This reduces the tendency of the network to converge on a local minimum in the training data instead of a more general solution. Training with added random noise is continued until a recognition goal (typically set at 95% or greater RMS) is reached.

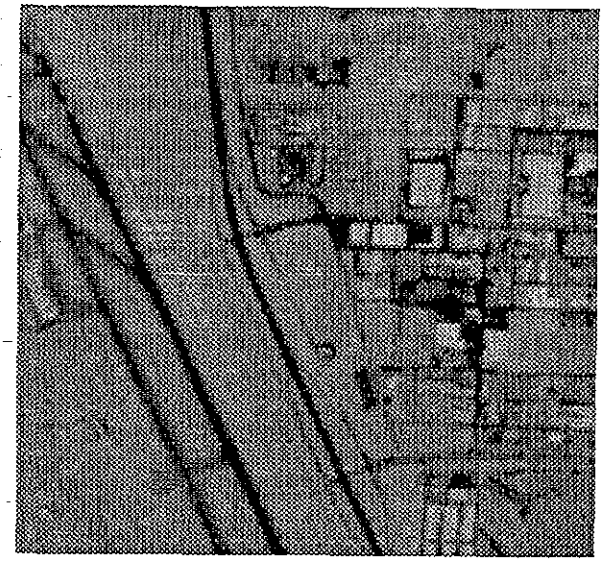

Figure 4: Sample phototexture

It is also beneficial to randomize the order of presentation in the training set, particularly where large numbers of identical, or nearly identical pixels are present in the exemplar set for a class. Without randomization, the network is easily trapped in a local minimum. This is indicated by tracking the identification error as each exemplar is presented to the network. An alternating pattern of high - low error as the exemplars are presented combined with an essentially fixed RMS error indicates a local minimum. Often, this is attacked by altering training parameters such as the percentage of random noise or reducing the training rate or "momentum" of the network. We have also found it useful to search the exemplars for identical pixels. Reducing the number of such

pixels can correct local minimum problems. The risk in this approach is in altering the distribution of pixels in an exemplar set such that it is no longer representative of the input image.

It is helpful to view the exemplar sets as occupying one or more regions in a space defined by the number of spectral bands available. For the three-band data used here, it can be called RGB-space, and represented as a simple volume. For LANDSAT Thematic Mapper data, this space would be a seven dimensional structure. In either case, the exemplar set for a single class may occupy several disconnected regions, with other classes occupying intervening regions. The complexity of the image data in this structure requires the use of a network with one or more hidden layers in order for the network to re-map the input data into the desired set of output classes.

The optimum number of hidden layer nodes is a subject of some debate. Too few hidden layer nodes will prevent the net from converging, while too many adversely effects performance. Image data as a rule is complex, with several discrete groups of pixels in RGB-space assignable to a single class. Water for example, may vary from bright blue through green to brown depending upon suspended sediment, regions of RGB-space which can also contain soil, vegetation, and man-made structures. The number of hidden layer nodes is equal, in the worst case, to the number of disconnected or meshed regions in the input distribution[5]. As this is often a difficult number to determine, rules-of-thumb have evolved. Lippmann[5] states that there must typically be more than three-times the number of hidden layer nodes as input nodes. For the three-band data used here, a number of hidden layer nodes equal to the product of the input and outputs nodes is a conservative starting point.

After training, the network is used to segment the entire image on a pixel by pixel basis. As shown in Figure 5, the segmented image pixel values are assigned to a single value depending upon the feature class. In practice, this is a useful point at which to examine the segmented image in comparison to the original. While scattered errors are expected, systematic errors can usually be traced to errors in constructing the training set. Often, this can be resolved by noting those regions of the image incorrectly classified, and adjusting the exemplars in the training set accordingly. For example, regions of vegetation in shadow may be mis-classified as asphalt. By ensuring that exemplars for shadowed vegetation are included in the training set, a successful segmentation often results.

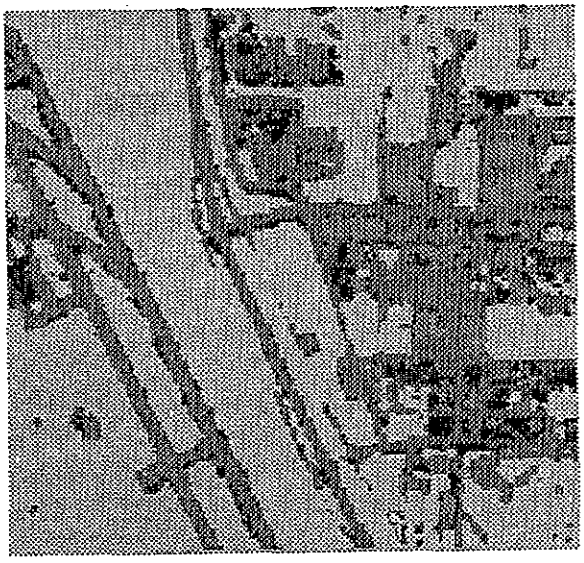

Figure 5: Example segmented image.

In other instances, the sets may not be separable. In our experience, beach sand is readily mis-classified as concrete. Close examination of the exemplar set values found that in the source image, many of the pixel values for the two surfaces are identical. This should not have been a surprise given that concrete contains a high percentage of local sand. A possible solution to this problem is obtaining source data with a larger number of spectral bands. To date however, we have elected to manually select and modify the critical mis-identified regions, which are readily detected in the segmented image. This is easily accomplished with commercial or public domain image manipulations tools.

While inseparable exemplars occur in natural imagery, a different problem exists in phototexture derived from visual system databases. In some cases, it may be desirable to use such phototexture as source data to avoid feature correlation issues. In such images, the total number of applied textures can be rather low, allowing nearly complete

characterization in the training set. Also, the textures are typically applied in geometric patterns obtained from DFAD data. Although selected to avoid visual discontinuities, the different textures may be readily classified by a well trained neural network, resulting in a patch-work appearance in the segmented image due to distinct texture boundaries. In such cases, adjusting the original phototexture pixel values based on the segmentation data may produce unwanted edges. Our second approach to generating a database (i.e., re-assigning pixels in a given segment to a defined range of random values based upon the feature class) allows this problem to be reduced. For example, assigning separable mean points, but overlapping distribution limits, to soil and vegetation classes allows large regions to remain distinct while minimizing border edges.

While the approach described is applied to radar images in our prototype system, the database generation process can be extended to other sensor systems as well. For example, the segmentation value assigned to a given pixel may be selected to represent an infrared reflectivity and material heat capacity rather than X-band radar reflectivity. In a similar manner, databases for MMMW radar can also be constructed. Using this approach, highly detailed, geospecific sensor simulation can be produced to support world-wide missions.

Real-time processing overview

Real-time processing centers on adding aspect sensitive radar features such as leading edge enhancement and far-shore brightening to the image, and adjusting the gamma of the resulting image to that typical of a sensor display as shown in Figure 6. These processes are implemented using image processing algorithms that are aspect sensitive, allowing the approximation of the required radar effects. Also in real-time, DTED data is ray-traced to find both shadowed areas and forward scattering slopes. This information is combined with the processed phototexture to adjust the displayed values of the pixels. Additional processing necessary for DBS and real-beam map displays is not depicted in the figure.

Figure 6: Simulated SAR image using adjusted phototexture. Simulated illumination direction is from the northwest.

A generic form of cultural shadowing is used to produce occulted areas behind structures for which actual elevation data is unavailable. Generating this entails performing a rear-edge detection on a structure-only segment of the data base, and extending this edge as a function of grazing angle. While the images generated should not be confused with a rigorous radar signature prediction, they do provide efficiently computed, effective sensor simulation for training.

COMPUTATIONAL RESOURCES

In the prototype DRLMS system and in subsequent developments, the number of computations required for the generation of an image is directly scaled by the displayed range and azimuth resolution of the simulated radar system. Since image processing to produce representative SAR-like images replaces the more rigorous calculations of traditional DRLMS, the total number of computations required in real time are greatly reduced. For example, a tactical radar simulation may display an effective 256 x 256 resolution cells (the size of the simulated SAR image shown in Figure 3). The processes applied require on the order of 100 floating point operations per displayed pixel to transform the pre-processed data base into a representative SAR image (plus any additional

overhead for the operating system, retrieving the database and performing interface processes). This translates to approximately 7 million floating point operations to build an image.

A modern RISC processor such as the MIPS R4000, operating at 100MHz, is rated at approximately 16 MFLOPS. Based upon the operations required, and neglecting the computing overhead previously mentioned, it is estimated that approximately 2 to 3 simulated SAR patches per second can be processed on a machine of this class. This is in good agreement with benchmark data from the Macintosh prototype, where processing times in the range of 15 seconds per patch are typical for a processor of 1/30th the rated floating point speed.

The benchmarks above predict the hosting of a real-time DRLMS on a single microprocessor. With newer generations of RISC processors such as the MIPS R4400 promising floating-point performance of 24 MFLOPS, further enhancements are achievable. This is in contrast to traditional DRLMS systems which depend upon special purpose hardware achieving hundreds of MFLOPS in parallel computing architectures.[6]

REFERENCES

1) Wehner, D.R. (1987) *High Resolution Radar* (PP 11-28) Norwood, MA: Artech House

2) Seldin, J.H & Cederquist, J.N. "Classification of Multispectral Data: A Comparison Between Neural Network and Classical Techniques" Proceedings of the Government Neural Network Applications Workshop: Volume 1 (PP 79-83) Wright-Patterson AFB, OH, 24-26 Aug. 1992.

3) Stengel, J.D. Jr. & Hoog, T.W. "DRLMS Technology- A Critical Assessment of the State-of-the-Art" Proceedings: 13th Interservice Training Systems and Education Conference (PP 35-43) Orlando, FL, 2-5 Dec., 1991.

4) Rumelhart, D.E. & McClelland, J.L. (1986) *Parallel Distributed Processing: Volume I: Foundations* (PP 318-362) Cambridge, MA: The MIT Press

5) Lippmann, R.P "An Introduction to Comuting with Neural Nets": IEEE ASSP Magazine (PP 4-22), April 1987.

6) Drew, E.W. & Matusof "Applying Advanced Parallel Processing Concepts to Radar Simulation and Image Generation" Proceedings: 15th Interservice Training Systems and Education Conference (PP 345-353) Orlando, FL, Nov. 29- Dec 2, 1993.