Increasing the a·c density by a reduction in bandwidth or an increase in number of w.ive numbers serves to reduce this image artifact.

V. Conclusions

We have investigated the relationship between object profile and received scattering data for the monostatic, or coincident source-receiver, tomographic measurement geometry. Use of this particular geometry in connection with field studies employing ground penetrating radar may lead to a more complete exploitation of acquired data. With this possibility in mind, we have tried to develop inversion schemes that (at least for two dimensional objects) might be implemented in the field using a minimal amount of computing power (e.g., a personal computer). We assume from the outset that the objects to be imaged are weak scatters so that the received data is linearly related to the object profile via (4). Direct numerical inversion of this equation is still, however, computationally quite demanding. For two-dimensional inhomogeneities, we find that by assuming the object to be deep we can relate the Fourier transform of the object profile, $\hat{O}$, to the Fourier transform of the data as in the Generalized Slice Theorem. For objects which are in the far field of the source (this can be achieved by appropriate data gathering strategies) further simplification is possible.

We have classified the derived inversion schemes as Fourier transform and far field methods, and have found that the Fourier transform method yields superior image quality. Since the implementation of such a method can be based on fast Fourier transforms, it offers considerable computational advantage over the far field method. While, in principle, application of the Fourier transform method is limited to relatively deep targets, numerical simulations suggest that there is only minor loss of image quality associated with the application of this approach to shallow targets.

As a final point we consider the use of a boundary condition which ignores reflections from the measurement surface. By virtue of this boundary condition, algorithms presented here do not account for multiples. For isolated inhomogeneities in a monostatic measurement geometry, multiples should not be significant since the measurement surface is not a specular reflector with respect to inhomogeneities with the possible exception of a small number of measurement positions. Multiples will be of concern when horizontal layering is present. In such situations, multiples can be suppressed by appropriate signal processing [9] prior to imaging.

References

[1] A.J. Devaney, "A filtered backpropagation algorithm for diffraction tomography." *Ultrasonic Imaging,* vol. 4, pp. 336–350, 1982.

[2] A.J. Devaney, "Geophysical diffraction tomography,*IEEE Trans. Geosci. Remote Sensing,* vol. GE-22, pp. 3–13, 1984.

[3] P.E.A. Fisk, "Strength of materials in and around sinkholes by *in situ* geophysical testing," in *Proceedings 2nd Multidisciplinary Conference on Sinkholes and Environmental Impacts of Karst,* pp. 153–156, 1987.

[4] R.W.P. King and S. Prasad,*Fundamental Electromagnetic Theory and Applications.* Englewood Cliffs, Prentice-Hall, NJ, 1986.

[5] W. Magnus, F. Oberhettinger, and R.P. Soni,*Formulas and Theorems for the Special Functions of Mathematical Physics.* New York: Springer-Verlag, 1966.

[6] P.M. Morse and H. Feshback, *Methods of Theoretical Physics, Part I.* New York: McGraw-Hill, 1953.

[7] G.J. Schneider and W.R. Peters, "Location of an underground opening with an acoustic cross borehole logging system and tomographic signal processing," in*Proceedings of the 3rd Technical Symposium on Tunnel Detection*, pp. 200–210, 1988.

[8] W.A. Schneider, "Integral formulation for migration in two and three dimensions,"*Geophysics,* vol. 43, pp. 49–76, 1978.

[9] A.B. Weglein and R.H. Stolt, *The Wave Physics of Downward Continuation, Wavelet Estimation, and Volume and Surface Scattering: II. Approaches to Linear and Nonlinear Migration-Inversion.* Philadelphia: SIAM, 1992.

[10] A.J. Witten and W.C. King, "Acoustic imaging of subsurface features," *Journal of the Environmental Engineering Division of ASCE,* vol. 116, pp. 116–181, 1990.

[11] A.J. Witten and W.C. King, "Sounding out buried waste." *Civil Engineering,* vol. 60, pp. 62–64, May 1990.

Terrain Classification in SAR Images Using Principal Components Analysis and Neural Networks

M.R. Azimi-Sadjadi, S. Ghaloum, and R. Zoughi

Abstract—**The development of a neural network-based classifier for classifying three distinct scenes (urban, park and water) from several polarized SAR images of San Francisco Bay area is discussed. The principal component (PC) scheme or Karhunen–Loeve (KL) transform is used to extract the salient features of the input data, and to reduce the dimensionality of the feature space prior to the application to the neural networks. Employing PC scheme along with polarized images used in this study, led to substantial improvements in the classification rates when compared with previous studies. When a combined polarization architecture is used the classification rate for water, urban and park areas improved to 100%, 98.7%, and 96.1%, respectively.**

I. Introduction

The Importance of high resolution ground mapping is growing in airborne and spaceborne radars. Synthetic-Aperture Radar (SAR) systems take advantage of the motion of a platform carrying a radar to synthesize the effect of a large antenna aperture rendering a high resolution image in the along-track direction [1]. One of the primary application of high resolution imagery using a SAR system is in terrain classification in areas such as ecology, agriculture, commercial, and military [2]. This is accomplished by assigning a color or a gray level value to each pixel of the major terrain (e.g., urban areas, parks, bodies of water, farmland, etc.) classes found in an image. Thus, when the classification process is completed, the radar image is transformed into a color-coded image showing several types of terrains. For example, a pilot who is forced into emergency landing does not have the time to read arrays of radar vectors and make a split second decision on where to land. On the other hand, if he has a color-coded image displayed in his cockpit, all he would have to do is to look for the color designating the park areas and direct his aircraft to those spots.

In this paper terrain classification is performed using artificial neural networks [3], [4]. The system consists of a preprocessor network to extract the features directly from the data and a classifier network to perform real-time terrain classification. The preprocessor stage consists of a single layer neural network trained with the constrained Hebbian learning algorithm [5], [6] to extract the principal components (PC's) or the Karhunen-Loeve (KL) transform [7] coefficients

Manuscript received January 21, 1992; revised August 27, 1992.

The authors are with the Electrical Engineering Department, Colorado State University, Fort Collins, CO 80523.

IEEE Log Number 9206399.

of the image data. This scheme not only reduces the dimension of the classifier weight space and extracts the salient features of the image, but also decorrelates the data which is an important property for training and testing the neural network classifier. The classification is performed by using a multilayer back-propagation network which provides the complex decision region needed for classifying different terrains in SAR images. Several images obtained using different radar polarizations are used to train the classifier network optimally. Comparison of the results of this paper with those of other similar studies [8] indicates substantial improvements in the classification rates.

II. Image Data

When an electromagnetic wave is reflected from an object or scene its polarization depends upon the polarization of the incident wave and the structure of the reflecting objects [1]. Thus, the polarization properties of a radar signal may be used to enhance discrimination among different terrains. Furthermore, by varying the polarization of the receiving antenna to different angles between 0° to 90° (in the plane transverse to the direction of propagation, for example if 0° is vertical polarization, 90° is horizontal polarization) the information extracted from a scene will be more complete. Namely, the polarization of the receiving antenna may be made to match that of the reflected wave. This is particularly useful as the polarization of the reflected wave can not always be predicted. Consequently, varying polarization angle should enhance the classification rate in SAR images. This can be achieved by employing a fully polarized wave (polarimetric) which can be expressed as the superposition of two linearly orthogonally polarized components, namely horizontally and vertically polarized components. In such case, there is enough information in the Stokes matrix (or scattering matrix) to synthesize any polarization configuration [9].

In 1985, a polarimeter operating at L-band was flown on a NASA aircraft and recorded data from the San Francisco Bay area which included three major scenes: urban areas, parks, and large bodies of water. This data was subsequently processed and stored at the Jet Propulsion Laboratory (JPL). This image required a large storage space. As a result, the data was reduced into a compressed format representing the Stokes scattering matrix of this image with a data compression of 12.8:1. Since the compressed data was not quantized and could not be used on image display systems, a software (MULTITEST) and a quantizer were needed to convert the compressed data to a gray level image [9]. In this study an artificial neural network is used to classify the three major scenes/terrains in this image.

III. Principal Components Method

Owing to the huge amount of SAR imagery data involved it is necessary to apply a compression technique to reduce the volume of data prior to the application to neural network classifier. This provides data compression for both training and testing data sets, and reduces the dimensionality of the classifier network weight space as well. In addition, training and testing based on the raw data (gray level values) could contain many redundancies and nonessential information which may lead to a lengthy training process.

The image data compression by means of PC extraction is accomplished by projecting each sample block of the data along the directions of the individual orthonormal eigenvectors of the covariance matrix of the image data [7]. If the first few eigenvalues of the covariance matrix contain most of the signal energy, the dimensionality of the data can be greatly reduced without losing much information by only retaining those components along with the principal eigenvectors. In addition to its optimality in data reduction, which is not shared by other feature extraction schemes, it provides salient features of the data that are decorrelated. This property is ideally suited for classification purposes as these decorrelated features or components can be used to train the classifier effectively.

The conventional approach for PC extraction involves the computation of the input data covariance matrix and then the application of a diagonalization procedure to extract the eigenvalues and the corresponding eigenvectors [7]. For large data sets, the dimensions of the covariance matrix grow significantly large hence making its computation and manipulation practically inefficient and inaccurate due to truncation/round-off errors. In addition, all the eigenvalues and their corresponding eigenvectors have to be evaluated even though only the eigenvectors which correspond to the most significant eigenvalues are used in the data transformation process. These deficiencies make the conventional scheme inefficient for real-time applications. Thus, to perform PC extraction efficiently, a method which evaluates the most significant eigenvectors of the data covariance matrix without the need to form this matrix is required. This can be accomplished using a dedicated neural network architecture trained with the constrained Hebbian learning rule [5], [6]. This algorithm updates the weights of linear neurons sequentially until all the weights converge to the desired eigenvectors and the outputs approach to the PC's of the data. This scheme is particularly useful for large size windows (blocks) since no covariance matrix is needed. In the following section this neural network-based scheme will be used for real-time PC extraction from SAR images.

A. PC Extraction Using Constrained Hebbian Learning Algorithm [5], [6]

The structure consists of an auto-association network which is trained directly from the data. The training is accomplished sequentially for each neuron using the constrained Hebbian learning rule [5], [6]. Each neuron receives a set of $L = N^2$ scalar input values $x_1(t)$, $x_2(t), \cdots, x_L(t)$, at the training sample t where the elements x_i's are the pixel values contained in each NXN block. These inputs are assumed to be the components of a stationary random vector process $X(t)$. The inputs are passed through a set of weights $w_{m1}(t), w_{m2}(t), \cdots, w_{mL}(t)$ to generate the output of the mth neuron, $y_m(t)$. There are $K \ll L$ output neurons, which upon completion of the training, extract the relevant eigenvectors of the data covariance matrix in their weight vectors and generate the associated PC's in their outputs. The input/output relation for the mth output neuron is given by

$$y_m(t) = W_m^t(t)X(t), \quad \forall\, m \in [1, K] \tag{13}$$

where $W_m(t)$ and $X(t)$ represent the weight and the input vectors respectively defined by

$$W_m(t) = [w_{m1}(t) \quad w_{m2}(t) \quad \cdots \quad w_{mL}(t)]^t \tag{2a}$$

$$X(t) = [x_1(t) \quad x_2(t) \quad \cdots \quad x_L(t)]^t \tag{2b}$$

where superscript t signifies the transposition operation.

To extract the first PC, the weight vector, $W_1(t)$, associated with the first neuron is updated based on the constrained Hebbian adaptation scheme using the following general learning equation

$$W_1(t+1) = \frac{W_1(t) + \gamma y_1(t)X(t)}{\|W_1(t) + \gamma y_1(t)X(t)\|} \tag{3}$$

where γ is a reasonably small positive scalar [5] and $\|\cdots\|$ represents the Euclidean norm of the relevant vector. The form of the denominator is due to the required orthonormal property of the weight vector.

Assuming γ is small, (3) can be expressed as a power series in γ

$$W_1(t+1) = W_1(t) + \gamma y_1(t)[X(t) - y_1(t)W_1(t)] \qquad (4)$$

where all the higher powers of γ have been neglected. The term $\gamma y_1(t)X(t)$ in (4) represents the Hebbian increment. On the other hand, the difference between the updated weight and the old weight is $\gamma y_1(t)[X(t) - y_1(t)W_1(t)]$, where $X(t) - y_1(t)W_1(t)$ is considered to be the effective input. The weight vector $W_1(t)$ tends to grow according to the input $X(t)$, but the growth is controlled by an internal feedback, $(-y_1(t)W_1(t))$, in the neuron. Since (4) is directly derived from (3), for small values of γ, the weight vector is kept reasonably close to one even when the higher order terms are neglected. It has been shown [5], [6] that the weight vector of the neuron would converge to the normalized eigenvector associated with the most significant eigenvalues of the data covariance matrix. In order to get the second most significant eigenvector, the contribution of the previously computed most significant component is subtracted from the data and the outcome is used to train another neuron using exactly the same learning equation. In principle, this is similar to Gram-Schmidt orthogonalization process [6]. In general, to extract the mth PC, assuming that all the previous m−1 neurons are trained, a new (mth) neuron will be trained using the following deflated input data

$$\hat{X}_m(t) = X(t) - \sum_{i=1}^{m-1} y_i(t)\hat{W}_i \qquad (5)$$

where $\hat{W}_i$ is the estimate of ith eigenvector and $y_i(t)$ is the corresponding output after convergence is reached. Performing (1), (4), and (5) sequentially, results in all the significant eigenvectors at the weights and PC's at the output of the neurons.

In addition to the significant reduction in the input data size, the other benefits of PC method include rotation and translation independence and data decorrelation which can improve the training of the neural network.

IV. Neural Network Classifier

The image used in our simulations is the San Francisco Bay SAR image shown in Fig. 1. The goal is to train a multilayer back-propagation neural network to classify three different terrains in this image namely water, park and urban areas. The training data set for the neural network consisted of 8×8 blocks of the image data chosen from each separate class and arranged in row-ordered form, i.e., by stacking rows together to give a vector of size $L = 64$. Both the raw input vectors and the preprocessed inputs using the PC scheme were used. In the latter case, the size of the input vector was reduced to only $K = 16$. The input data was normalized in both cases so that the nodal values would not get too large during the training process. The architecture for the most parts was either a three-layer or a two-layer network determined empirically. Three output neurons were used to indicate three distinct classes. The desired outputs for urban, park and water classes were chosen to be 000, 111, and 010, respectively.

During the training process an input vector from each class was sequentially presented to the neural networks and the weights were updated such that the actual outputs were as close as possible to the specified desired outputs in the mean squared error sense. The weight adjustment was performed using the gradient descent method in conjunction with the error back-propagation scheme [2]. This process was repeated several times for all the training vectors until the average mean squared error (AMSE) over all the training samples and all the output neurons reached a prescribed minimum. The necessary number of iterations to reach the convergence is greatly dependent on the complexity of the error surface and the step size chosen for adaptation.

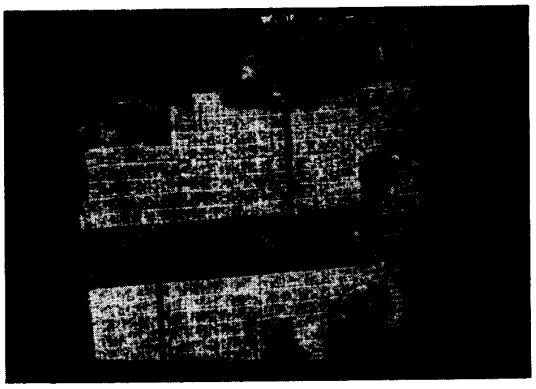

Fig. 1. Original SAR image of San Francisco Bay (polarization 30°−30°).

A. Training and Testing Using Raw Data

Total of 100 training input vectors from each class were used to train the neural network. An appropriate network architecture consisted of 64 inputs and 27, 9 and 3 neurons in the first hidden, second hidden, and output layers, respectively was found empirically. After the completion of the training (1000 iterations), the generalization capability of the network was tested on a testing set consisting of the rest of the image data. The results are given in Table I for different polarizations of the radar. Polarization of 30°−30°, for instance, means that the transmitting and the receiving antennas have polarization angles of 30°. Note that the classification rates for 0°−0° polarization are higher than those of 90°−90° polarization. This can be seen by comparison of the corresponding images where distinction between urban and park classes in 0°−0° polarized SAR image is slightly easier than that in 90°−90° polarized image. The results of 0°−90° polarized image, which are not included, were very low because the network was not capable of separating the two classes (urban and park). The reason being in this case the polarization loss factor which is proportional to the cosine of the angle between the respective polarization vectors is zero or in practice very small. As a result, the reflected power retrieved by the radar is very small which in turn implies that the information conveyed is not sufficient for differentiating different classes.

B. Training and Testing Using PC Data

In this experiment, the PC of each input vector containing 64 pixel values was taken and the first $K = 16$ PC's were extracted using the neural network-based approach with the constrained Hebbian learning algorithm. The PC data was then presented to the neural network classifier. The terrain classification using both two-layer and three-layer network was tested. Due to the fact that the input data contained independent features no improvement in classification rates was observed from the added computational power of the three-layer neural network. As a result, a two-layer network consisting of 16 inputs, 7, and 3 neurons in the hidden and output layers was used. To be consistent with the training scheme in the previous section, 1000 iterations were taken to train the network. After the training was completed, the generalization capability of the network was tested utilizing all the testing data vectors for each class. The reduction in the size of the input vectors resulted in considerable reduction in the size of the network which in turn led to faster training process and slightly better generalization capability as evident from the results in Table I.

To improve the performance of the overall system even further, we combined the outputs of the four neural networks corresponding to

TABLE I
CLASSIFICATION RATES FOR VARIOUS POLARIZATIONS USING THE RAW TRAINING DATA.

Polarization (deg - deg)	*Urban* Raw %	*Urban* PC %	*Park* Raw %	*Park* PC %	*Water* Raw %	*Water* PC %
0 - 0	97.5	97.5	89.5	89.0	100.0	100.0
30 - 30	96.5	98.0	90.5	91.0	100.0	100.0
45 - 45	95.5	97.0	90.5	90.0	100.0	100.0
60 - 60	90.5	94.0	88.0	87.0	100.0	100.0
90 - 90	89.0	89.0	82.5	83.5	99.0	99.5

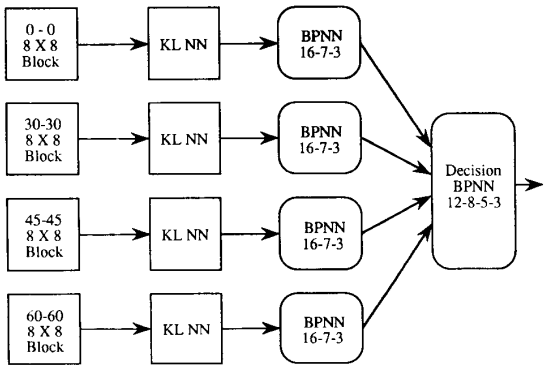

Fig. 2. Schematic diagram of the classifier using combined polarization architecture.

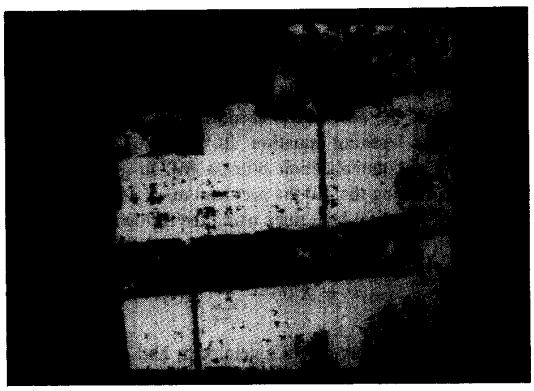

Fig. 3. Color coded SAR image (polarization 30°−30°).

0°−0°, 30°−30°, 45°−45°, and 60°−60° polarizations by another network which performs the decision task, as shown in Fig. 2. By presenting the outputs of the networks corresponding to different polarized images to this decision network, the classification rates can further be improved. In other words, the classifier "looks" at each 8 × 8 block with different polarization angles before making a final decision. As was explained earlier, different polarizations emphasize different classes, hence resulting in bias in the decision of the classifier. This bias can be avoided when combined polarization are applied to the composite network. Combining various polarizations also leads to improvements in the generalization capability of the system when new terrains are encountered. The decision neural network used to combine the outputs of the different classifiers consisted of 12 inputs, 8, 5, and 3 neurons in the first hidden, second hidden and output layers, respectively. Using this scheme the classification rates were improved to 98.7%, 96.1%, and 100% for urban, park, and water areas, respectively.

To generate the color coded SAR image, a moving window of size 8 × 8 was swept across the entire image. The PC's were then extracted from each window and fed to the trained network. Each one of the three classes was assigned a pixel value. Therefore, the output of the network was one of these three assigned pixel values. As the window moved across the image, the output of the neural network was assigned to the four pixels in the center of the block. This was to speed up the classification procedure since the window was shifted by two pixels at a time instead of only one. The other reason for assigning four pixels at a time is to avoid small variations within a class (e.g., swimming pools in urban areas and small buildings in park areas). The color-coded image using the neural network-based KL transform is shown in Fig. 3 (for polarization 30°−30°). This image which consists of only three gray levels illustrates the three distinct classes.

V. CONCLUSIONS AND DISCUSSIONS

Based upon the results of previous section, the following observations can be made. When the raw data was used as inputs to the neural network, a relatively large three-layer network was needed to perform the classification task. However, when the PC data was used, the size of the network reduced to a two-layer (16-7-3) as a consequence of energy compaction property of this method. By having smaller neural networks, the convergence time for updating the weights was significantly reduced. Furthermore, the classification rates of the network when PC method was used showed to be higher than when no preprocessing (raw data) was employed. This improvement is the result of the ability of PC method to extract the uncorrelated components of the data. Comparison between the classification rates achieved in this work and those of [8] shows substantial improvements owing to the use of PC scheme and different polarization images. Classification rates were further improved when a combined multipolarization network was used.

In conclusion, the results in this paper indicate that neural networks indeed perform very well when used to classify different terrains in SAR images. The major advantage of neural network classifiers over the conventional statistical pattern classification/recognition schemes is that no statistical assumption about the distribution of SAR data is needed as they are nonparametric in nature and can learn the underlying distribution of the data during the training process. In addition, unlike the conventional statistical classifiers [2], they can closely capture the signal distributions generated by nonlinear and/or non-Gaussian processes. Neural networks generally offer considerable improvements in classification rates in comparison to the conventional statistical methods. It is shown in this paper that this improvement in the performance is particularly substantial when these networks are used in conjunction with a preprocessing scheme such as the

PC method. Finally, the generalization capability of these networks make them ideal for pattern classification/recognition under varying environments.

REFERENCES

[1] F. T. Ulaby, A. K. Fung, and R. K. Moore, *Microwave Remote Sensing, Passive and Active, Vol. II.* Dedham, MA: Artech House, 1986, ch. 9.

[2] J. A. Kong, S. H. Yueh, H. H. Lim, R. T. Shin, and J. J. Van Zyl, "Classification of earth terrain using polarimetric synthetic aperture radar images," in *Progress in Electromagnetics Research, Vol. III, "Polarimetric Remote Sensing"*, New York: Elsevier, 1990.

[3] J. A. Freeman and D. M. Skapura, *Neural Networks: Algorithms, Applications and Programming Techniques.* Addison-Wesley, 1991.

[4] S. Ghaloum, "Terrain classification in SAR images using principal components analysis and neural networks," M.S. Thesis, Colorado State University, Fall 1991.

[5] E. Oja, "A simplified neuron model as a principal component analyzer," *Journal of Math Biology*, pp. 267–273, 1982.

[6] T. D. Sanger "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, vol. 2, pp. 453–475, 1989.

[7] A. K. Jain, Fundamentals of Digital Image Processing. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[8] S. E. Decatur, "Application of neural networks to terrain classification," *Proc. of the International Joint Conference on Neural Networks, Vol. I*, pp. 283–288, June 18–22, 1989.

[9] T. Thompson, *et al.*, "NASA JPL Aircraft SAR Compression Tape Information Package," JPL, Pasadena, CA, April 1987.

The SSABLE System: Automated Archive, Catalog, Browse and Distribution of Satellite Data in Near-Real Time

James J. Simpson and Daniel N. Harkins

***Abstract*— Historically, locating and browsing satellite data has been a cumbersome and expensive process. This has impeded the efficient and effective use of satellite data in the geosciences. SSABLE is a new interactive tool for the archive, browse, order, and distribution of satellite date based upon X Window, high bandwidth networks, and digital image rendering techniques. SSABLE provides for automatically constructing relational database queries to archived image datasets based on time, data, geographical location, and other selection criteria. SSABLE also provides a visual representation of the selected archived data for viewing on the user's X terminal. SSABLE is a near real-time system; for example, data are added to SSABLE's database within 10 min after capture. SSABLE is network and machine independent; it will run identically on any machine which satisfies the following three requirements: 1) has a bit-mapped display (monochrome or greater); 2) is running the X Window system; and 3) is on a network directly reachable by the SSABLE system. SSABLE has been evaluated at over 100 international sites. Network response time in the United States and Canada varies between 4 and 7 s for browse image updates; reported transmission times to Europe and Australia typically are 20–25 s.**

Manuscript received March 20, 1992; revised September 1, 1992. This work was sponsored by the Marine Life Research Group (MLRG) of the SIO and by grants from NSF, NASA, and ONR. NSF funded development of SSABLE. ONR provided funds for hardware. NASA provided funds for rearchiving historical data.

The authors are with the Scripps Satellite Oceanography Center, University of California, San Diego, La Jolla, CA 92093.

IEEE Log Number 9206929.

I. INTRODUCTION

Satellite data are large, complex data sets acquired sequentially in time. Sequences of images derived from these data can provide a basis for the detection of global change processes [2]. The construction of these sequences, in turn, benefits greatly from the use of database management and geographical information systems (e.g., [6], [10]). Given the size and generally cumbersome nature of historical satellite data archives, it is imperative that efficient and effective systems for the archive, catalog, browse, and distribution of remotely sensed data be developed, if spacecraft data are to play a major role in Global Change studies (e.g., [2]).

Inexpensive yet powerful UNIX-based workstations now are generally available. They use a network/graphical protocol called X Window and can be linked to satellite data centers via national networks. In addition, more compact, robust archival media (e.g., DAT tape) now are available. Exploitation of these newer technologies provided the basis for the Scripps Satellite Archive and Browse for Localized Environments (SSABLE) system, an interactive tool for the archive, browse, order, and distribution of satellite data in near real-time.

II. CONCEPTUAL DESIGN OF THE SSABLE SYSTEM

A. Overview of Design Constraints

The SSABLE System was designed to serve two distinct types of users: 1) satellite data centers; and 2) scientific and/or operational end users. Data centers acquire and archive satellite data, update catalogs and browse files, and distribute products to a broad user community. They also support experiments in near-real time. Generally, users need to efficiently search, browse, and order selected satellite scenes from a large database, based upon a variety of search criteria.

Additional constraints placed on the design of SSABLE were: 1) device independence; 2) extensibility; 3) intuitive graphical user interface; 4) compact storage and longevity of archival media; 5) reduced costs; 6) on-line, data quality control; 7) on-line audit control of data; 8) near-real time applications support; and 9) inexpensive, rapid response, global access.

Based on these criteria, tools were chosen to implement SSABLE. They include an open, industry standard software architecture (UNIX, C, X Window), a Graphical User Interface (GUI) based on the OSF/Motif widget set, a Relational Database Management System implemented in the C programming language and the Structured Query Language (SQL), nonproprietary network protocols (TCP/IP), use of national networks (e.g., Internet), and better archival media.

B. Hardware Configuration

SSABLE uses three networked workstations. The Deepdish system controls satellite data acquisition. It consists of a Scientific Atlanta 5 meter steerable antenna, antenna controller, bit and frame synchronizers and a multiple frequency receiver interfaced to a Hewlett–Packard HP9000/370 workstation which receives satellite data transmitted in either the L or S band. Data is captured using the Global Imaging System 9000 Data Capture software but SSABLE does not require a specific data capture module. After capture, SSABLE initiates the archive, catalog and browse generation process on Deepdish. The Oddie system performs archive, catalog, browse file creation, and updating for newly acquired and historical data. Oddie also services all requests (e.g., browse, order, accounting) from users distributed