Naval Research Laboratory
Stennis Space Center, MS 39529-5004

NRL/MR/7240-93-7026

Knowledge-Based Oceanographic Image Compression

RONALD J. HOLYER
JUANITA R. CHASE
Remote Sensing Applications Branch
Remote Sensing Division

March 1993

REPORT DOCUMENTATION PAGE

Form Approved
OBM No. 0704-0188

1. Agency Use Only *(Leave blank).*	2. Report Date.	3. Report Type and Dates Covered.
	March 1993	Final

4. Title and Subtitle.

Knowledge-Based Oceanographic Image Compression

5. Funding Numbers.

Program Element No. 0603207N
Project No. 0514
Task No.
Accession No. DN252024
Work Unit No. 72519803

6. Author(s).

Ronald J. Holyer and Juanita R. Chase

7. Performing Organization Name(s) and Address(es).

Naval Research Laboratory
Remote Sensing Applications Branch
Stennis Space Center, MS 39529-5004

8. Performing Organization Report Number.

Memorandum Report 7026

9. Sponsoring/Monitoring Agency Name(s) and Address(es).

Space and Naval Warfare Systems Command
2451 Crystal Drive
Arlington, VA 22202

10. Sponsoring/Monitoring Agency Report Number.

Memorandum Report 7026

11. Supplementary Notes.

12a. Distribution/Availability Statement.

Approved for public release; distribution is unlimited.

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*

The Navy has a requirement for image compression on the order of 30:1 for transmission of satellite imagery to the Fleet. Many common image compression algorithms will not retain the desired image fidelity at this compression ratio. This study investigates the use of domain specific oceanographic knowledge in the compression process as one means of achieving the required performance. A transform-based compression algorithm, which draws upon a large, representative image archive for a statistical description of oceanographic knowledge, is proposed and demonstrated. Very high compression ratios seem to be achievable by this method (much greater than 30:1), but the exact evaluation of performance potential requires further study. Among the unique characteristics of this approach is the ability to fill in cloudy areas with estimated image features. While knowledge-based image compression shows much promise for very high compression ratios, it does have significant drawbacks. The most significant of these are the requirement for a large image archive and much computation in order to build the region specific, knowledge-based algorithm. For these reasons, the compression approach demonstrated here is not seen as the single solution to all of the Navy's satellite image compression problems. However, in many applications the proposed algorithm could be very beneficial. In concert with other more generic image compression algorithms, knowledge-based image compression should play an important role.

14. Subject Terms.	15. Number of Pages.
Remote Sensing, Image Compression	29
	16. Price Code.

17. Security Classification	18. Security Classification of Report.	19. Security Classification of This Page.	20. Limitation of Abstract. of Abstract.
Unclassified	Unclassified	Unclassified	SAR

KNOWLEDGE-BASED OCEANOGRAPHIC IMAGE COMPRESSION

1.0 INTRODUCTION

The Navy has a requirement to transmit satellite images from receiving sites to regional centers and shipboard units where detailed local analyses are performed. In view of the high data volume associated with satellite imagery and the relatively small bandwidth available to environmental products on Navy communication channels, some form of image compression prior to transmission is required. A recent analysis of anticipated image volume and Navy communication facilities indicates a 30:1 compression ratio will be required to solve the Navy's satellite communication problem.

Most image compression algorithms are designed to be generic, i.e., to give acceptable performance on as many types of imagery as possible. The generic approach is logical in applications where the type of imagery encountered will vary widely. However, there are some applications where the nature, content, and/or statistical characteristics of the imagery are well known and very limited. In these applications, the compression algorithm can be designed to take advantage of the limited scope and known properties of the imagery, thereby producing better compressions than those produced by generic algorithms. This report describes a study conducted to demonstrate one possible method of incorporating oceanographic knowledge into an application specific compression algorithm for the Gulf Stream region of the North Atlantic. The images considered are infrared(IR) (i.e., brightness proportional to sea surface temperature(SST)) images from the National Oceanic and Atmospheric Administration (NOAA) Advanced Very High Resolution Radiometer (AVHRR) sensor. Comparable images are produced by the Optical Line Scanner (OLS) on the Defense Meteorological Satellite Program (DMSP) satellites. Conclusions drawn here with respect to the AVHRR data would also apply to the OLS.

2.0 IMAGE COMPRESSION OVERVIEW

Data compression is perhaps the fundamental expression of Information Theory, a branch of mathematics arising in the late 1940s with the work of Claude Shannon at Bell Labs. Data compression (in this case image compression) is possible because many common schemes for encoding imagery contain redundancy. Redundant information in an image results in unnecessary bits being added to the image file size. Redundancy can be of three forms, spatial, spectral, and temporal. This study deals with single banded images, so spectral redundancy does not apply, although it is often the most important when considering multispectral data sets. Spatial redundancy refers to the fact that the value of a

pixel is correlated to some degree with the values of neighboring pixels. Temporal redundancy refers to the fact that today's observation of an ocean characteristic at some location is correlated to some degree with the value of that observable at the same location yesterday. By encoding each pixel with a fixed number of bits, pixels are treated as independent observations -- an inefficient, redundant scheme since it does not take advantage of correlation of pixel values with the values of spatial and temporal neighbors.

Information Theory uses the concept of entropy as a measure of how much information is encoded in an image. The higher the entropy, the more information the image contains. The entropy of a pixel (in units of bits of information) is defined as the negative logarithm of the probability of the pixel having the value that it does (Nelson 1991).

Number of bits = -log(probability of the recorded brightness) (1)

The entropy of the entire image is the average number of bits per pixel over all pixels in the image. If the entropy of an image, in average bits per pixel, is less than the number of bits per pixel used to encode the image, then some degree of compression is possible.

Entropy defines the upper limit on the compressibility of an image. One can get infinitely close to the entropy value but cannot exceed it without losing information. For the AVHRR thermal image archive used in this study, the entropy was calculated to be 6.97 bits per pixel. The raw images here are encoded with 10 bits per pixel. By reducing the dynamic range of the data to exclude cloud temperatures, these images have been reduced to 8 bits. Some additional lossless compression (to 6.97 bits per pixel) is theoretically possible when treating the image data as a sequence of random numbers with known probabilities. One way to improve the compression ratio is to take advantage of spatial/temporal correlation that exists in the data set. Introduction of spatial/temporal correlation is one way of utilizing oceanographic knowledge to improve image compression for a specific type of imagery (at the loss of general applicability to other types of imagery).

A simple way to incorporate spatial correlation into the compression algorithm is to encode a pixel as a difference between its value and that of its neighbor. If spatial correlation is high, then the difference values will be predominately 0, 1, 2, and other small numbers. Small numbers can be encoded with fewer bits than larger numbers, so the neighbor difference image should be more compressible than the original image. This experiment was performed with the image data set assembled for this project. The entropy of the neighbor difference images was found to be 4.19 bits per pixel. Neighbor difference images could, therefore, be compressed with a ratio of 2.39:1 with respect to the original 10-bit image. Better ways of utilizing spatial correlation could undoubtedly be developed, and temporal correlation could also be

considered. However, even with these improvements, the best lossless compression ratio for these oceanographic IR images of the Gulf Stream is likely to be on the order of 3:1. Thus, it is apparent that lossless compression is not likely to satisfy the Navy's 30:1 compression ratio requirement.

If one allows information loss in the compression/expansion process, then much higher degrees of compression are possible. Numerous techniques for lossy compression have been developed. With any technique, the limit to the amount of compression is generally the amount and type of loss that is acceptable for a given application. This study consists of an investigation of lossy compression by transform techniques. Transform techniques model an image as a weighted sum of "basis" images. Any image can then be represented by the weights rather than by an array of pixels. If the number of weights required to achieve an acceptable reconstruction take less storage space than the original image, then some degree of compression has been accomplished. The assumption here is that the basis images have been prestored or can be generated at the receiving site.

In transform-based compression the basis images are generally constructed from some mathematical function. For example, the discrete cosine transform (DCT) uses the two-dimensional cosine function to form basis images. Figure 1 shows the DCT basis images for an 8 x 8 pixel image block. The mathematical regularity and x,y symmetry of the DCT basis images are apparent from Fig. 1. The proposed Joint Photographic Experts Group (JPEG) standard for image compression calls for partitioning of an image into 8 x 8 blocks with each block transformed independently using the basis images shown in Fig. 1. The DCT algorithm has many advantages or it would not have been selected as the JPEG standard. However, for compressing oceanographic images for Navy applications, the DCT has two disadvantages. First, transformation of the blocks independently can result in block-boundary artifacts that appear in the reconstructed imagery. Secondly, the method does not take advantage of oceanographic knowledge.

Oceanographic knowledge can be incorporated into transform-based image compression by utilizing oceanographic information in the formulation of the basis images, such that the basis is optimized for the known characteristics of the set of images being compressed. Sections 4 and 5 describe the transform-based compression of a satellite IR image of the Gulf Stream Region. Our transform-based compression results are encouraging, but work has not proceeded to a point where a reliable estimate of possible compression ratios is available. Section 7 describes additional work that will be required to further define the useful limits of this approach.

3.0 DATA SET

NOAA-AVHRR thermal images of the Gulf Stream were selected as

a test data set. In order to minimize costs for this first feasibility demonstration, only previously processed images (available in the Naval Research Laboratory (NRL) archives) were considered. Previous projects such as the GEOSAT Ocean Applications Project (GOAP) and the 6.2 Data Assimilation Technology project have produced many earth located IR images of the Gulf Stream Region. Forty-two of the "best" of these previous images were selected for this study. Best in this context means most cloud-free. The images covered the years 1985 through 1989. All months of the year except January and February were represented at least once in the data set. However, the data was predominately from April, May, and June, which are typically the most cloud-free months in this region. All images were maximum temperature composites of several passes. The compositing procedure further reduced cloud contamination.

Test images are warped to a Mercator projection with the upper-left corner corresponding to 45^{0}N, 75^{0}W. Images were 1024 x 1024 pixels with each pixel representing an area of approximately 2.5 km on each side. (Note that this is not the full resolution of the AVHRR instrument, which is 1.1 km.) Figure 2 is an image that is representative of the set of 42. Warmer temperature values are represented by darker shades in the images, and colder temperatures are represented by lighter shades.

Since this feasibility demonstration is limited to the computing power of a Sun SPARC2 workstation, the first step in data processing was to eliminate the bottom half of the images leaving 1024 x 512 pixels. The discarded bottom half of the images covered the Sargasso Sea area, which is frequently cloud covered and normally isothermal. This generally featureless area presents no particular challenge to image compression algorithms, so there seemed to be no reason to lengthen the computations by retaining this area.

3.1 Residual Cloud Removal

Even though the original images were relatively cloud-free data and cloud effects were further reduced by compositing several passes, the residual cloudiness in the test images was judged to be too prominent to be ignored in the calculation of basis images for compression. To further reduce cloud effects, each composite image was compared pixel-by-pixel with the Generalized Digital Environmental Model (GDEM) climatology of surface temperatures. If the difference between climatology and the image exceeded some threshold, then the pixel was judged to contain a cloud. The image-climatology difference threshold could be adjusted interactively while viewing the image on the screen. The analyst adjusted the threshold until the largest number of residual cloud pixels was flagged as clouds without flagging too much of the image that was obviously not cloud covered. When the cloudy pixels had been identified in this manner, all those flagged as cloudy were replaced with the climatological SST values plus some offset value. The offsets from climatology for cloud replacement were also varied interactively so that the analyst could "blend" the cloudy areas

with the cloud-free areas. Figure 3 is the same image as Fig. 2 with the southern half cropped and the interactive climatological cloud detection and replacement completed.

GDEM climatology is available for each month. In this study, only four monthly climatologies were used. The February climatology was used for cloud screening and cloud replacement in images from January, February, and March. The April climatology was used for images from March, April, and May. Likewise, climatologies for July and October completed the year.

3.2 Image Subsampling

To further reduce the computational load for this study, the resolution of the test images was reduced by subsampling by a factor of 16 in each direction. The resultant spatial resolution increased from 2.5 to 40 km, and the images were reduced to 64 x 32 pixels. The upper-left corner, which contained land, was also cropped from the image. The final image then contained a total of 1848 pixels, which was still enough to stretch the computational capabilities of the SPARC2. Figure 4 is the same image as Figs. 2 and 3 with the upper-left corner cropped and the resolution reduced to 40 km.

4.0 CALCULATION OF BASIS IMAGES

4.1 Covariance Matrix

It is computationally convenient to consider images to be vectors (one-dimensional arrays) rather than two-dimensional arrays. Images (64 x 32 pixels) are converted to vectors by placing the 64 pixels from the first row into vector elements 1 through 64, the pixels from the second row of the image in vector elements 65 through 128 and so on for all 32 rows of the image. The vector then contains 64 x 32, or 2048 elements. (In the present case, the vector contained 1848 elements since clipping the land areas from the image resulted in less than 64 pixels in some rows.)

$$\vec{X} = (i_{1,1}, i_{1,2}, \ldots, i_{1,64}, i_{2,1}, i_{2,2}, \ldots, i_{32,64}), \qquad (2)$$

where $i_{m,n}$ is the intensity of image I at row m, column n.

The covariance matrix **C** for the data set is then given by

$$\boldsymbol{C} = E\{(\vec{X} - \vec{M})^T(\vec{X} - \vec{M})\} \qquad (3)$$

where $\vec{M}$ is the mean image vector and E{ } denotes expectation. The eigenvectors of **C** are calculated and then unpacked as illustrated below into image format to serve as basis images for compression.

$$B = \begin{bmatrix} x_1 & x_2 & \cdots & x_{64} \\ x_{65} & x_{66} & \cdots & x_{128} \\ & \vdots & & \\ x_{1984} & x_{1985} & \cdots & x_{2048} \end{bmatrix} \tag{4}$$

where x_i is the i^{th} element of $\vec{X}$.

Eigenvector calculations on **C** were attempted on a SUN SPARC2 using the C language routines TRED2, TQLI, and EIGSRT from Press *et al.* (1986). These routines did not produce valid eigenvectors in this case. Eigenvectors were visually "bad" as they were interspersed with black holes. Mathematically the eigenvectors were also suspect because they were highly nonorthogonal. (By definition eigenvectors must be orthogonal.) The reason for the failure of these well-known eigen routines has not been determined. However, this does draw attention to one of the problem areas associated with the present approach to image compression. The calculation of eigenvectors of extremely large matrices is not a straightforward problem. Indeed, eigenvectors of large matrices is in itself a research area in the field of scientific computing. It was decided to implement an eigenvector algorithm found in the work of Simonds (1963), due to the above mentioned problems.

The eigenvectors of **C** transformed back into image format by Eq. (4) form the desired set of basis images for image compression. These basis images contain oceanographic information as a result of their origin in the covariance matrix, which was calculated from an archive of oceanographic images. Specifically, the oceanographic knowledge contained in the basis images is a statistical description of the spatial covariability of SST in the Gulf Stream Region. The 20 largest eigenvalues of **C** are given in Table 1. The cumulative variance column of the table shows that a weighted sum of the first 20 eigenvectors will account for 98.4% of the total variance of the test image data set. This is an encouraging result for image compression where the concentration of variance in a small number of eigenvectors indicates a potentially high degree of compression. The eigenvectors associated with the 40 largest eigenvalues were saved and unpacked into image form to serve as the basis vectors for image compression. Figures 5-9 show the first 5 of these basis images. Note that the basis images shown contain Gulf Stream and eddy-like structure rather than the regular geometrical patterns of the DCT basis images shown in Fig. 1. Since Figures 5-9 do contain "ocean features" they are somewhat optimized for compression of Gulf Stream images as was the objective of this study.

5.0 IMAGE COMPRESSION

If there were no clouds in the image, I, then the weighting coefficient, w^k, for the k^{th} basis image, B^k, would be the sum of the products resulting from a pixel-by-pixel multiplication of I by B^k.

$$w^k = \sum_m \sum_n i_{m,n} * b^k_{m,n} \qquad (5)$$

However, when part of the image is cloud covered, it is not desirable to include the cloud brightness in the calculation of compression weights. To eliminate the cloudy areas from consideration in the compression process, a cloud mask was generated from the image to be compressed. The cloud mask contained "0" to indicate cloud and "1" to indicate no cloud. A new cloud-free basis image, B^k_{cf}, is then formed by a logical AND operation between B^k and the cloud mask.

$$B^k_{cf} = B^k \; AND \; mask \qquad (6)$$

The cloud-free basis image, B^k_{cf}, is then substituted for B^k in Eq. (5). The result is the compression weight for the k^{th} basis image. The original image can be reconstructed (approximately) from the basis images and these weights. If the basis images are prestored at the receiving site, then only the weights must be transmitted in order to recreate the image at the remote site. If the number of weights are less than the number of pixels in the image, then some degree of image compression has been achieved.

In this study, the weights have been encoded as 16-bit signed integers resulting in 16 bits of transmitted message for each basis image required in the reconstruction. The calculated compression weights for Fig. 2 are given in Table 2. These 40 weights, at 16 bits each, result in a total message length of 640 bits. The original image of the area covered by Fig. 2 contains approximately 2560 samples, 1280 lines, and 10 bits per sample for a total message length of 32,768,000 bits. The 640 bits used to encode Fig. 2, therefore, represent a compression ratio of 51,200:1.

6.0 IMAGE RECONSTRUCTION

Using the transmitted compression weights from Table 2 and the prestored basis images such as those shown in Figs. 5-9, the receiving site can reconstruct a corrupted version of the original image. The reconstruction of image I, I_r, is just a weighted sum of the basis images.

$$I_r = \textit{Mean Image} + \sum_k w^k * B^k \qquad (7)$$

Figure 10 shows the reconstructed version of Fig. 4. Note that many of the major mesoscale features present in the original can be identified in the 52,100:1 compression. The root-mean-square (rms) temperature difference between Figs. 4 and 10 is 2.24°C.

7.0 DISCUSSION AND RECOMMENDATIONS

Several features of this compression algorithm are noteworthy. First, the heavy computational burden is in the formation of the basis images. The actual compression and reconstruction procedures involve only image products and image sums. These are simple operations that can be performed rapidly even by low-end image processing systems. The computationally demanding basis image formation is done only once for a given geographic area, and it can be done at a central site where Cray or Connection Machine power is available.

Secondly, basis images must be prestored at both the transmitting and receiving sites. The storage requirement could be significant, depending on the image lossiness that is permitted, the size of the area, and the resolution of the imagery. Worst case storage requirements could be as much as 800 Gb. The typical storage requirements for a Navy Op Area might be on the order of 100 Mb, which is easily manageable.

Thirdly, the basis images contain no cloud patterns (because we replaced clouds with climatology before calculating basis images). Also, the compression weights are calculated based on only the cloud-free portions of the image. Therefore, cloud data is not reflected in the compression weights. Lacking any information on cloud patterns from basis images or weights, the reconstructed image contains SSTs throughout. The cloudy areas have been filled in with estimates of SST values. Estimates are based on SST patterns observed in the cloud free areas combined with "knowledge" of typical SST patterns contained in the basis images, via their origin in the covariance matrix. This feature would not be possible with a generic compression algorithm.

If a satellite image sent to a remote site was to be used to initialize a numerical model, and if that image contained clouds, then optimal interpolation (OI) would be needed at the remote site to assign estimated SST values to the cloud covered grid points. However, if the cloud contaminated image had been compressed for transmission using the algorithm considered here, then the OI would have already been performed as a by-product of the image compression. Computing time to perform OI at the remote site would be eliminated. Of course, some users might not want their data set to contain large areas of estimated values. In these cases, a run

length encoded binary cloud mask could be efficiently transmitted to permit the end user to identify the areas of cloud fill-in.

The useful compression ratio for the method presented here has not been determined because (a) this work was done at reduced resolution as a result of the limited computational power of the Sun SPARC2, and (b) only 41 images were included in the test data set. The obvious extension of the work then is to move to the Naval Oceanographic Office (NAVOCEANO) Cray-YMP computer system or to the NRL Connection Machine and repeat the analysis at higher resolution. The repeated experiment must also include expansion of the satellite image data set used in calculating the covariance matrix.

We believe that upon further investigation it will be found that compression ratios on the order of 1000:1 may be possible with this method. Because of the potentially outstanding performance, knowledge-based image compression should not be ignored. However, the excellent performance is not without cost. The cost in this case is the effort to collect the required archive of representative satellite imagery and to "build" the basis images. For standard Navy Op Areas where images are transmitted daily over many years, the investment to build basis images is probably wise since once the basis images are built, they can be used to provide extremely efficient compression for many years. However, for small volumes of unique images the present method has no value because the effort to build the basis images would not be justified. The knowledge-based image compression is, therefore, not seen as the one compression algorithm that will solve all Navy satellite image compression problems, but in combination with generic algorithms such as DCT or fractal transforms, the knowledge-based approach should be valuable. Further evaluation and development of this method is, therefore, recommended.

What has been accomplished here is a simple demonstration of the method, development of some approaches for dealing with cloud cover, experience with difficulties associated with the method (notably, machine precision in calculating eigenvectors of large matrices), and development of software tools. These accomplishments position us to further investigate knowledge-based image compression methods in the future.

To determine the useful compression ratio implies a standard for measuring usefulness. Such a standard is not obvious, and indeed establishment of such a standard should be the subject of further research. The mean squared error (MSE) between the original and the reconstructed image is perhaps the most widely used measure of compression usefulness. However, MSE may not be the most meaningful metric for the present type of imagery. For example, the compression algorithm could do very poorly in cloudy areas but very well in cloud-free areas. The poor performance over clouds would result in a large MSE. The large MSE would indicate poor performance when in fact the clouds are of no concern and the performance over the areas of interest is excellent.

One could conceive of numerous metrics for judging performance of a compression algorithm. Some examples are listed below.

Distribution of Error One can hypothesize that errors around gradients in the image are not as important as errors in uniform regions. The basis for this hypothesis is that any good edge detector will still find the edges, even if there is significant data loss there. What one doesn't want is to introduce variability (i.e., edges) where the original image is uniform. Therefore, one can tolerate much less reconstruction error in uniform regions than near gradients. If this hypothesis is true, then one might want to examine error jointly with gradient information. Under this approach, a metric could be developed that weighted MSE calculations with gradient values.

Topological Fidelity One common method of low level image segmentation is to assign a topological label, such as peak, valley, saddle point, etc., to each local neighborhood in the image (Haralick *et al.* 1983). This segmentation approach is presently utilized in some of the automated image interpretation work at NRL. An appropriate metric for judging lossy compression might then be the percent of image neighborhoods that change their topological classification as a result of the compression.

Derived Products Rather than judge compression based on the raw image data itself, it could be useful to calculate a MSE value based on some geophysical parameter derived from the image. As an example consider Multichannel Sea Surface Temperature (MCSST). MCSST is calculated from the difference between two spectral channels of the AVHRR sensor. For example, a typical MCSST algorithm is

$$MCSST = 0.983T_4 + 2.6616(T_4 - T_5) - 276.96 \qquad (8)$$

where T_4 and T_5 are the observed SSTs in AVHRR channels 4 and 5, respectively. Here accurate MCSST values from compressed images will depend on the preservation of the difference in apparent SSTs observed in the two different spectral bands. Note in Eq. (8) the temperature difference term $(T_4 - T_5)$ is multiplied by a factor of 2.6616, which has the effect of amplifying any errors in the temperature difference that result from compression. In this case, it would seem reasonable to go ahead and calculate MCSST for the original and compressed data sets, and then calculate a MSE value for MCSST rather than the MSE values for each of the individual channels.

Several of the above ideas could be combined into a single metric. For example, all of the above concepts could be incorporated if one worked with a derived geophysical parameter, like MCSST, used changes in topological class within the MCSST array as the metric, but included only those class changes occurring in cloud-free areas of the image.

Ideas for metrics for evaluating image compression algorithms are numerous. The above metrics are just typical examples. The best metric for judging image compression will be dependent upon the intended application of the data. Development of performance metrics should be a vital part of future Navy image compression research.

8.0 ACKNOWLEDGMENTS

This work was sponsored by the Space and Naval Warfare Systems Command under Program Element 0603207N, CDR P. Ranelli, Program Manager. Mr. D. Minton of SPAWAR PMW 165 provided interest, guidance, and support.

9.0 REFERENCES

1. Nelson, M. (1991). *The Data Compression Book* (M&T Books, Redwood City, CA) 527 pp.

2. Press, W.H., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling (1986). *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, pp. 335-380.

3. Simonds, J.L. (1963). "Application of Characteristic Vector Analysis to Photographic and Optical Response Data," *J. of the Optical Society of America*, 53, 8, pp. 968-974.

4. Haralick, R.M., L.T. Watson, and T.J. Laffey (1983). "The Topographic Primal Sketch," *Int. J. Robotics Res*, 2, pp. 50-72.

5. Rabbani, M. and P.W. Jones (1991). "Digital Image Compression Techniques," *SPIE Optical Engineering Press*, 220 pp.

TABLE 1 – Eigenvalues of the Covariance Matrix, **C**.

Eigen Value Number	Eigen Value	Cumulative Variance (%)
1	3033079	88.6
2	86443	91.8
3	58398	92.8
4	35775	93.9
5	25331	94.6
6	15895	95.1
7	14419	95.5
8	12232	95.9
9	10608	96.2
10	9779	96.5
11	9075	96.7
12	8048	97.0
13	7878	97.2
14	7092	97.4
15	6732	97.6
16	6200	97.8
17	5958	97.9
18	5644	98.1
19	5401	98.3
20	5247	98.4

TABLE 2 – Compression weights for Fig. 2.

Basis Image Number	Compression Coefficient	Basis Image Number	Compression Coefficient
1	-986	21	-28
2	422	22	-3
3	315	23	-41
4	-209	24	-8
5	-67	25	-66
6	34	26	110
7	160	27	-18
8	62	28	24
9	-67	29	38
10	-40	30	-55
11	46	31	-28
12	41	32	54
13	-77	33	52
14	4	34	70
15	68	35	-41
16	-38	36	-63
17	-21	37	35
18	27	38	-3
19	52	39	-19
20	-50	40	65

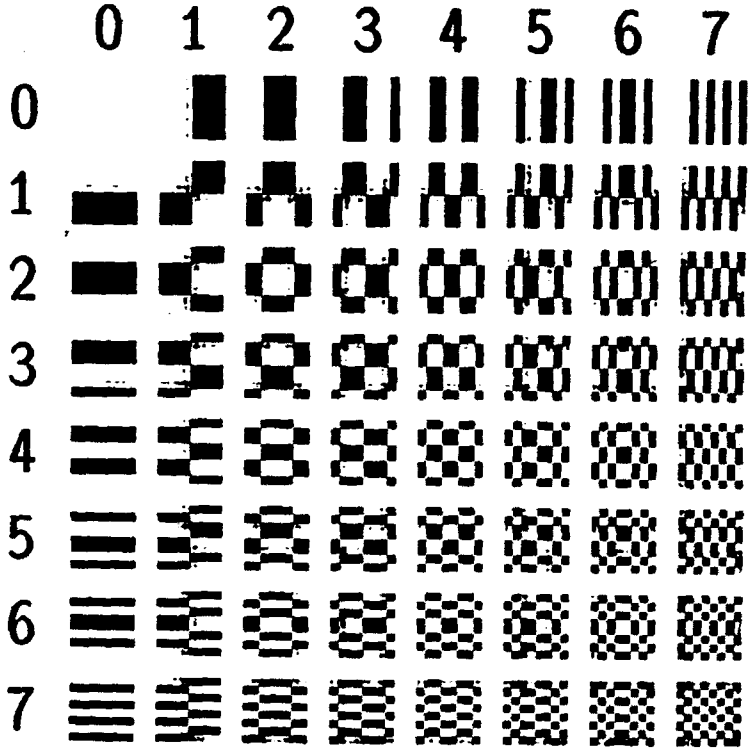

Fig. 1 – 8 x 8 DCT basis images (from Rabbani and Jones 1991).

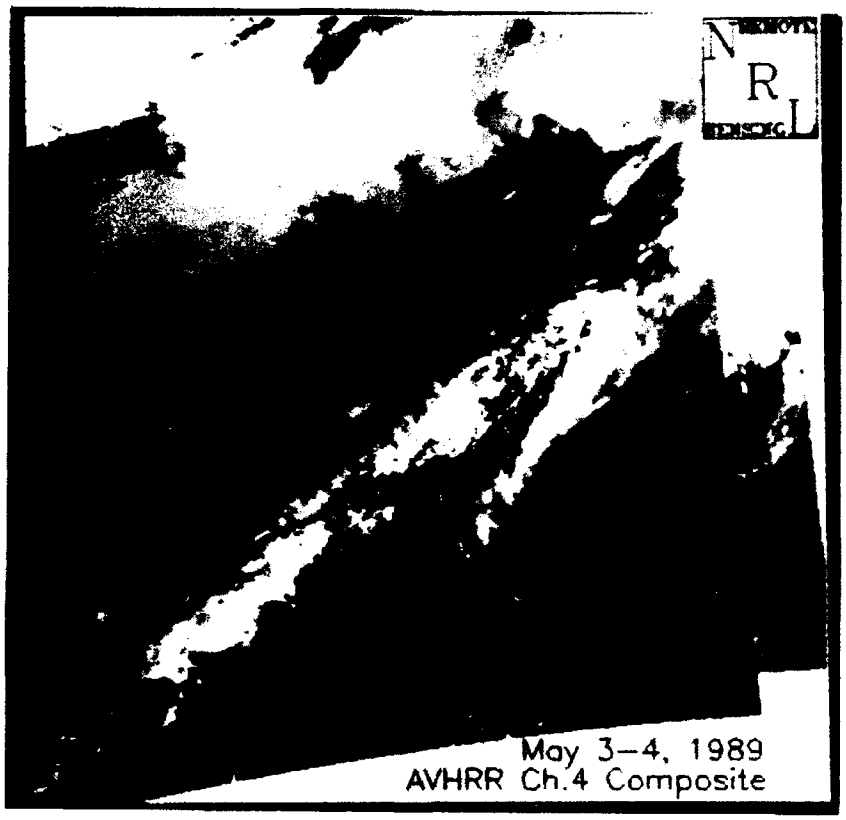

Fig. 2 - Typical NOAA Gulf Stream infrared image used in this study. A maximum temperature composite of several AVHRR channel 4 images from 3 and 4 May 1989.

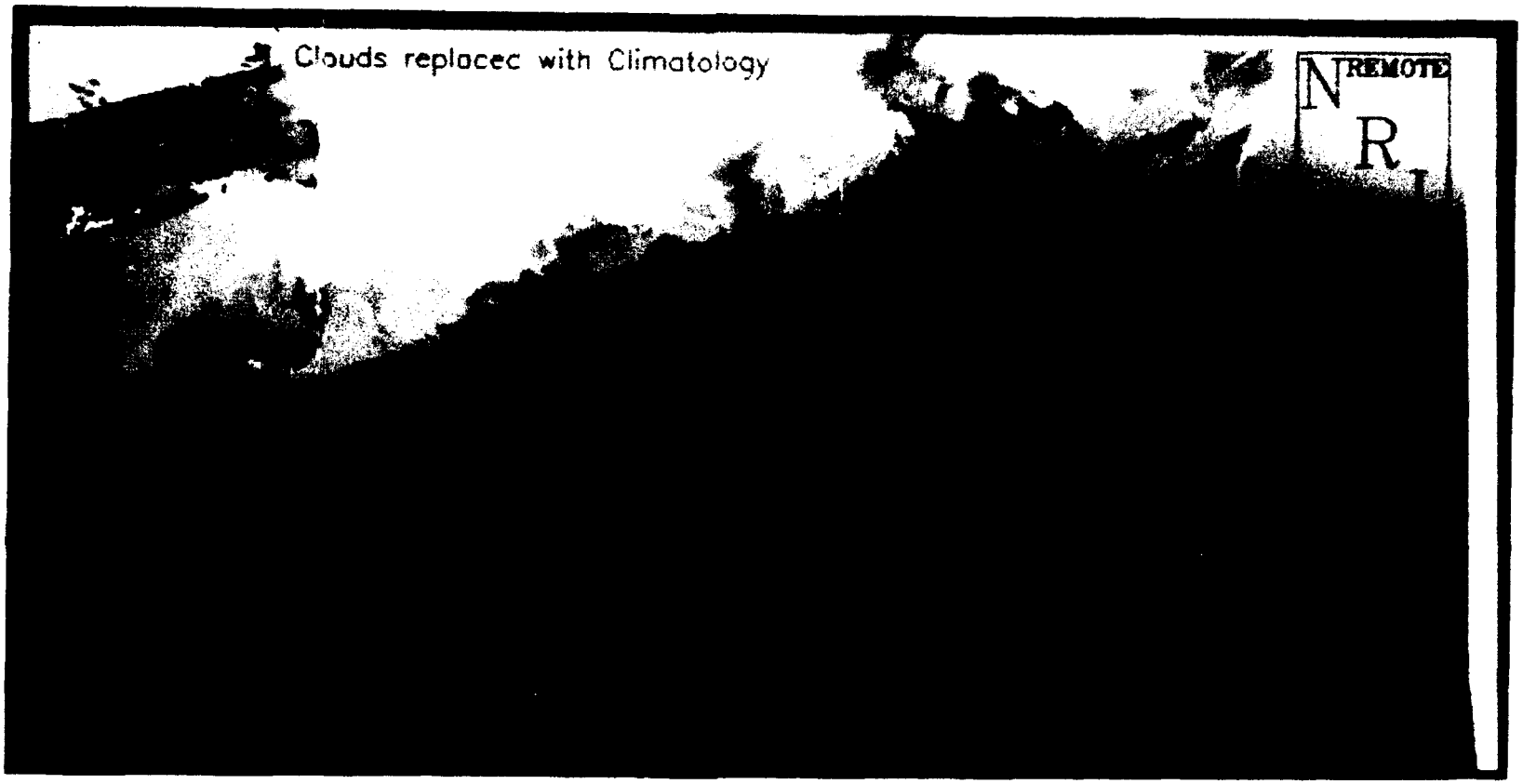

Fig. 3 - Image of Fig. 2 after cloudy pixels are replaced with climatology and Sargasso has been cropped.

Fig. 4 - Reduced resolution version of Fig. 3

Fig. 5 - Basis image number 1

Fig. 6 - Basis image number 2

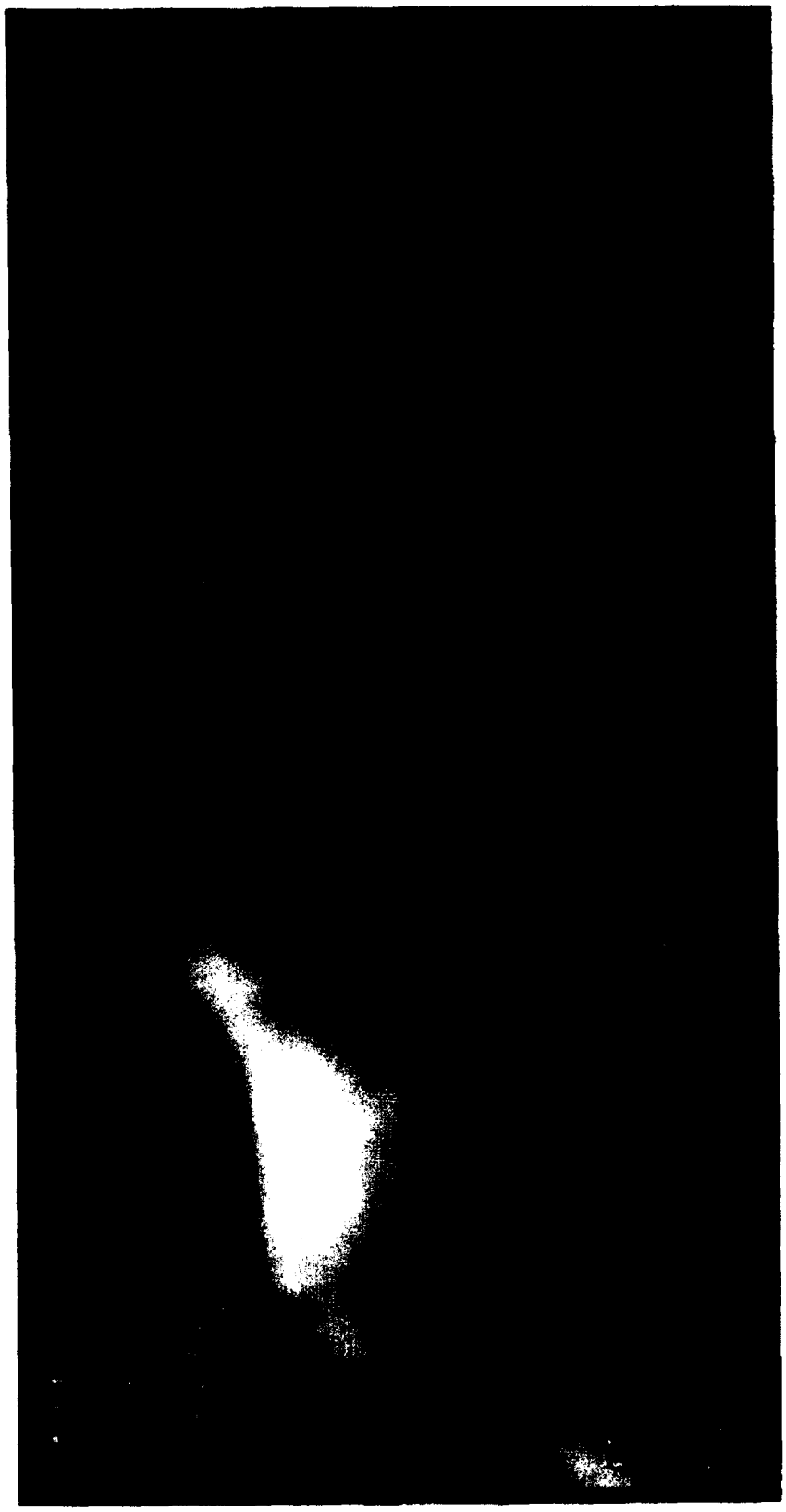

Fig. 7 - Basis image number 3

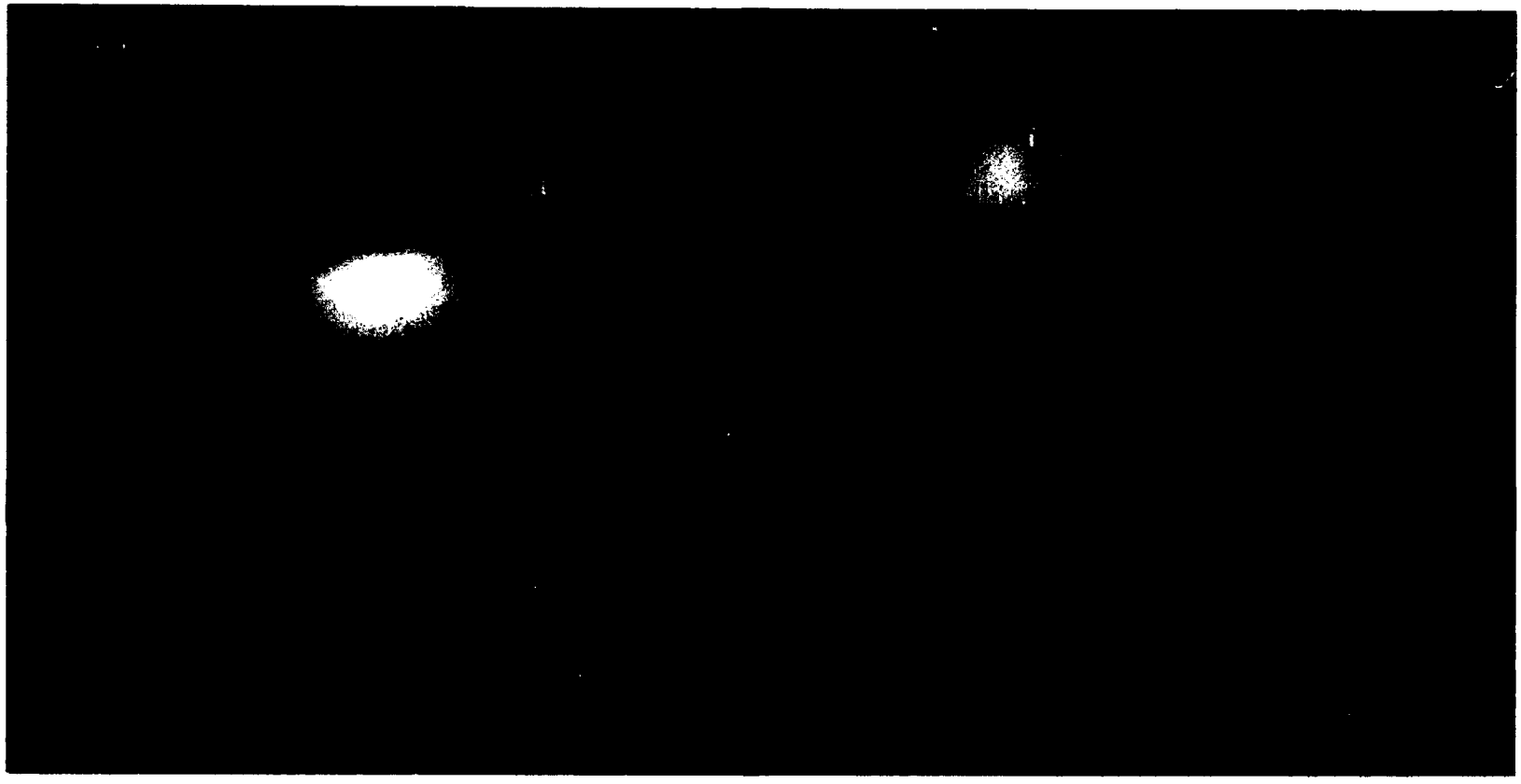

Fig. 8 - Basis image number 4

Fig. 9 - Basis image number 5

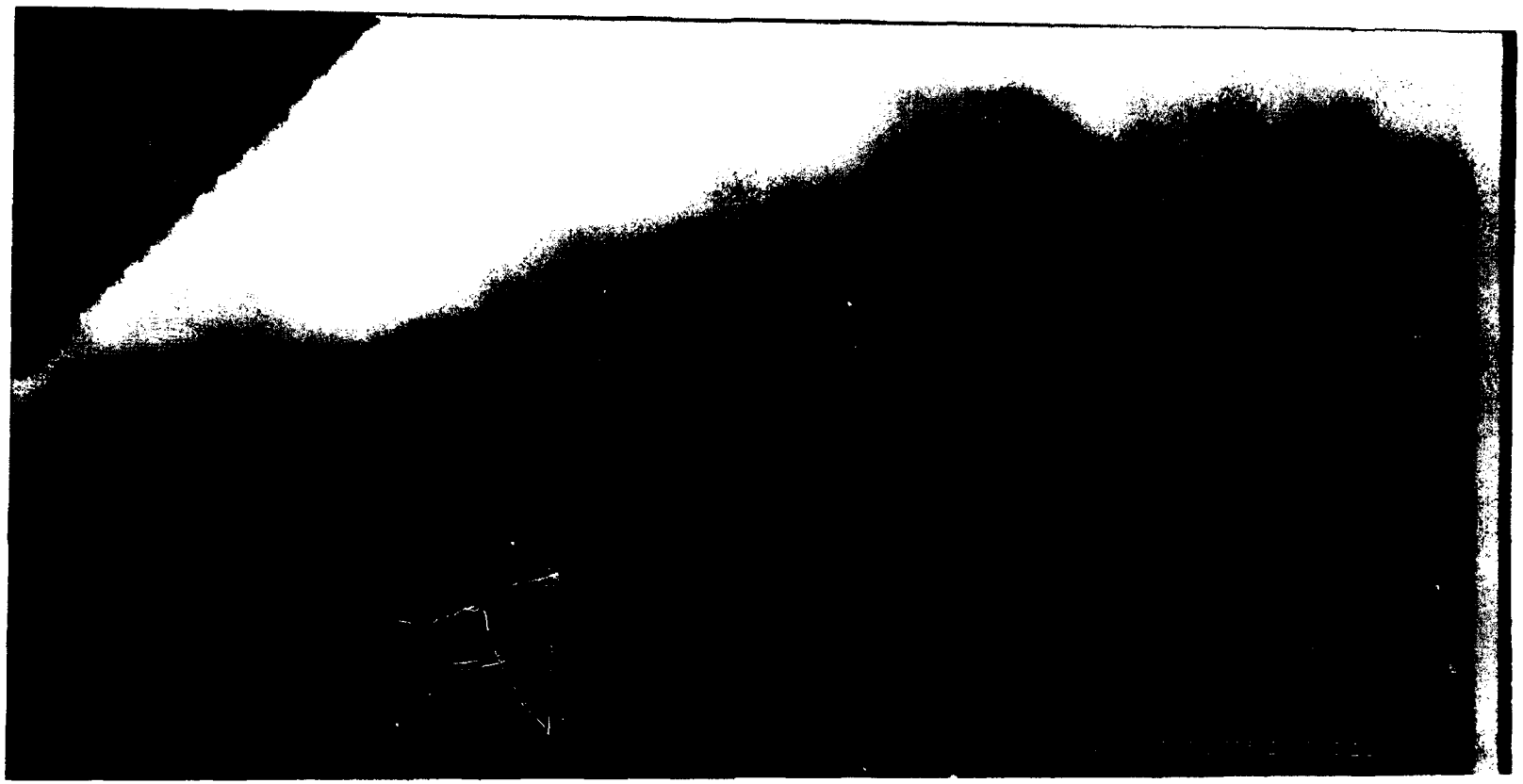

Fig. 10 - Reconstructed version of Fig. 4. Reconstruction is from a 51,200:1 compression.

APPENDIX A

PV-Wave program to calculate eigenvectors of large real, symmetric matrices. Program developed after procedures given in Simonds (1963).

```
;*****************************************************************
;
;                        Program COMEIGVVS.PRO
;
;*****************************************************************
;
;
;  This program creates eigenvalues and eigenvectors for a real,
;    symmetrical matrix.  The algorithm implemented here can be
;    found in Simonds (1963) work.
;
;
;  PARR contains the input data generated from VEC.PRO.
;  PPTUARR contains the input data PARR multiplied by UARRAY.
;  PPTUBARR contains the normalized values of PPTUARR. Normalized
;    by dividing PPTUARR by the largest element in PPTUARR.
;  UARRAY is an approximation of the eigenvector.  It
;    is initialized with ones.
;  UARRAY2 contains the squared elements of the PPTUARR array.
;  VARRAY contains a scaled version of PPTUARR. The scale value
;    is computed by taking the square root of the quotient of the
;    eigenvalue divided by the summation of the squared elements
;    of PPTUARR.
;  VVTARRAY contains the result of the VARRAY multiplied by its
;    transpose.
;  L1 contains the eigenvalue.
;  NVEC contains the eigenvector.
;  VSIZE is the size of the input data.
;  NVAL is the number of eigenvalues and eigenvectors to be
;    computed.
;  COUNTER counts how many eigenvalues and eigenvectors have been
;    computed.
;  RATIO contains the result of the ratio between the trace of
;    the input and the eigenvalue computed.
;
;
vsize = 1848
nval = 40

parr = dblarr(vsize,vsize)
vvtarray = dblarr(vsize,vsize)
uarray = dblarr(vsize,1)
uarray2 = dblarr(vsize,1)
varray = dblarr(vsize,1)
pptuarr = dblarr(vsize,1)
pptubarr = dblarr(vsize,1)
nvec = dblarr(vsize,1)

jday = ''
```

```
; Print date, time and read in input data

print,systime(jday)
print,'reading cpmat '

openr,1,'/sips3d1/chase/cpmat'
readu,1,parr
close,1

print,'finished reading cpmat '
print,parr(0,0),parr(vsize - 1,vsize - 1)

; Open output file to contain all the eigenvalues.

openw,3,'/sips3d1/chase/eigval_f26.dat'

counter = 0
ratio = 0

; Begin iterative procedure to compute eigenvalues and
;   eigenvectors.  In this case only the first 40 eigenvalues and
;   eigenvectors will be computed.

REPEAT begin

uarray(*) = 1.0

oldmax = 0.0
newmax = max(uarray)

; Repeat until the difference between the old max and the new max
;   is sufficiently small.

REPEAT begin

  oldmax = newmax
  pptuarr = parr # uarray

  newmax = max(pptuarr)

; Normalize PPTUARR by dividing each of its elements by its
;   largest element.

  for i = 0,vsize - 1 do begin
    pptubarr(i,0) = pptuarr(i,0) / newmax
  endfor

  if ((abs(newmax - oldmax) / newmax) ge 1.0e-5) then begin
    uarray = pptubarr
  endif

ENDREP UNTIL ((abs(newmax - oldmax) / newmax) lt 1.0e-5)

; The desired eigenvalue is the maximum value of the UARRAY in
;   the last iteration.
```

```
l1 = newmax
printf,3,l1,format='(f20.12)'
print,counter+1,l1

uarray2 = pptuarr * pptuarr

nvec = pptuarr / (sqrt(total(uarray2)))

; Write eigenvectors to individual files. One file for each
;   eigenvector.

filen = '/sips3d1/chase/eigdbl' + strtrim(string(counter+1),2) +
'.out'
openw,2,filen
  printf,2,nvec,format='(f20.12)'
  printf,2,''
close,2

scale = sqrt(l1 / total(uarray2))

varray = pptuarr * scale


; Compute the ratio of the eigenvalue to the trace of the input
;   to find what percent of the variability is explained by this
;   eigenvalue.

trace = 0.0
for i = 0,vsize - 1 do begin
  trace = trace + parr(i,i)
endfor

ratio = l1 / trace

vvtarray = varray # transpose(varray)

; Form new matrix that has all the roots and vectors of the
;   original except for the eigenvalue(s) and eigenvector(s) just
;   previously computed.

parr = parr - vvtarray

counter = counter + 1

ENDREP UNTIL (counter gt nval - 1)

close,3
print,systime(jday)
end
```

REFERENCE

1. Simonds, J.L. (1963). "Application of Characteristic Vector Analysis to Photographic and Optical Response Data," *J. of the Optical Society of America*, 53, 8, pp. 968-974.