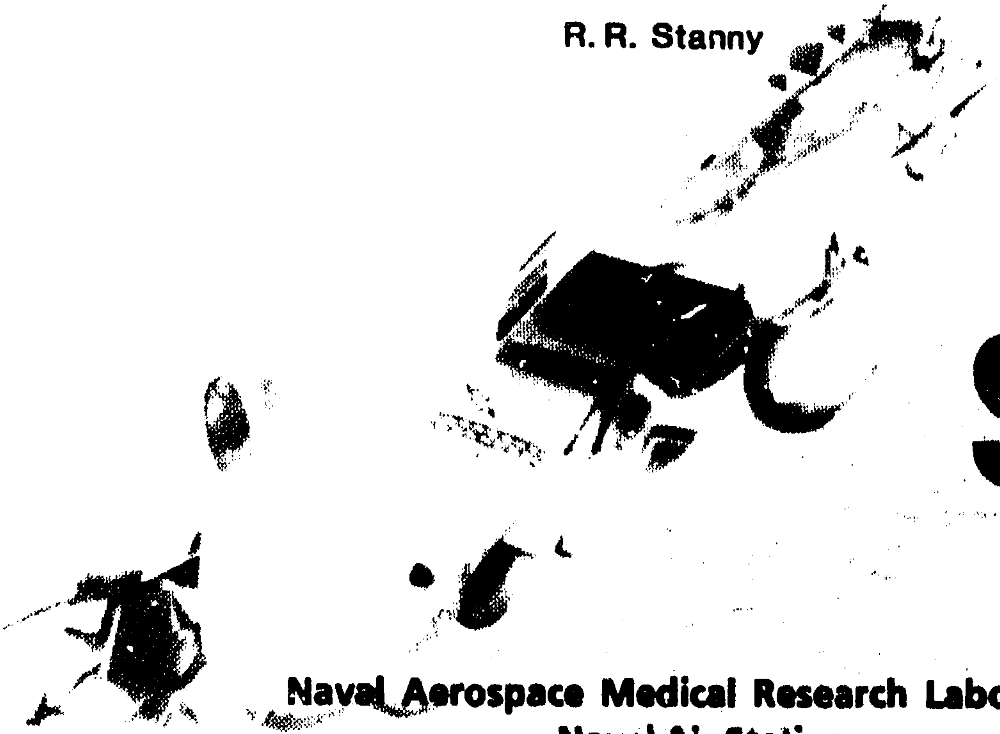
2

MAPPING THE EVENT RELATED POTENTIALS OF THE BRAIN: THEORETICAL ISSUES, TECHNICAL CONSIDERATIONS, AND COMPUTER PROGRAMS

R. R. Stanny



Naval Aerospace Medical Research Laboratory
Naval Air Station
Pensacola, Florida 32508-5700

Approved for public release; distribution unlimited.

Reviewed and approved 20 Oct 1988

J A Brady

J. A. BRADY, CAPT, MSC USN
Commanding Officer

DEPARTMENT OF THE NAVY ★ AEROSPACE MED RESEARCH LABORATORY ★

This research was sponsored by the Naval Medical Research and Development Command under work unit 61152N MR00001.001 7041.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
NAMRL SR 88-1

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Naval Aerospace Medical Research Laboratory

6b. OFFICE SYMBOL *(If applicable)*
21

7a. NAME OF MONITORING ORGANIZATION
Naval Medical Research and Development Command

6c. ADDRESS *(City, State, and ZIP Code)*
NAS, Pensacola, FL 32508-5700

7b. ADDRESS *(City, State, and ZIP Code)*
NMC, NCR, Bethesda, MD 21814-5044

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
61152N	MR00001	001	7041

11. TITLE *(Include Security Classification)* (U) Mapping the Event Related Potentials of the Brain: Theoretical Issues, Technical Considerations, and Computer Programs

12. PERSONAL AUTHOR(S)
R. R. Stanny

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM 1987 TO 1987

14. DATE OF REPORT *(Year, Month, Day)*
20 Oct 1988

15. PAGE COUNT
24

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Electroencephalography, Brain Mapping, Evoked Potentials, Event-Related Potentials

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This report discusses a variety of methods that have been used to generate spatial representations of the EEG. It deals with pertinent aspects of the recording process, as well as with the spatial analysis per se. Some of the limitations of the various approaches are described, and a general plan of implementation adaptable to most situations is outlined. The appendix contains two computer programs that can be used to generate surface-distribution maps of evoked responses.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
J. A. BRADY, CAPT MSC USN, Commanding Officer

22b. TELEPHONE *(Include Area Code)*
(904) 452-3286

22c. OFFICE SYMBOL
00

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603

SECURITY CLASSIFICATION OF THIS PAGE

DD Form 1473, JUN 86 (Reverse)

SECURITY CLASSIFICATION OF THIS PAGE

PREFACE

This report documents one laboratory's experience in developing a system for topographic studies of event-related potentials (ERPs) of the brain. It summarizes a number of practical and theoretical issues that arise when mapping and interpreting the surface distributions of ERPs. The author hopes the discussion of these issues will provide information useful to others who contemplate undertaking this type of research.

The report's introductory section briefly discusses the rationale for mapping the surface distributions of ERPs. Section 1 outlines a number of electrophysiological recording considerations pertinent to topographic studies. Section 2 deals with a number of practical considerations in the generation of surface maps. Section 3 describes a set of computer programs for studies of the surface distributions of human ERPs.

The Appendix contains source listings for the computer programs described in Section 3. These programs were developed for research in human ERPs. The programs are quite general, nonetheless, and could be used to generate displays of the electroencephalogram (EEG), the magnetoencephalogram (MEG), and other types of spatially organized data.

Many of the topics covered in this report are discussed in fairly concrete terms. That approach seems justified, given the existence of several good, general discussions (e.g., 1-5), and the degree of the author's contact with the mechanics of building a system of this type.

SUMMARY PAGE

THE PROBLEM

Event-related potentials (ERPs) are the summed membrane potentials of large numbers of simultaneously active neurons. To determine the anatomical location of the population of cells that produces a specific ERP, one must first record its responses from a number of points to characterize the spatial distribution of its surface field.

A number of factors affect the validity of spatial analyses of ERP phenomena. These include the number of recording electrodes used, the choice of the reference electrode, any distortions introduced when preprocessing the EEG, and any distortions introduced by the mapping process itself.

FINDINGS

This report discusses a variety of methods that have been used to generate spatial representations of the electroencephalogram. It deals with pertinent aspects of the recording process, as well as with the spatial analysis per se. Some limitations of the various approaches are described, and a general plan of implementation adaptable to most situations is outlined. The Appendix contains two computer programs that can be used to generate surface-distribution maps of ERPs.

Keywords: Brain mapping, Evoked potentials, Electroencephalography

ACKNOWLEDGMENTS

The author gratefully acknowledges the support of CAPT J. O. Houghton. This research was sponsored by the Naval Medical Research and Development Command under work unit 61152N MR000.01.001-7041.

INTRODUCTION

RATIONALE

The localization of function in the brain is an old problem that is important in evaluating neurological disease, environmental insult, and factors affecting human performance in general. Several noninvasive techniques exist for assaying function in the human brain. Of these, evoked-response electroencephalography and magnetoencephalography are unusual in that they can produce successive observations at intervals of milliseconds, the time scale on which neuronal responding occurs. Hence, they can produce information about the sequencing of brain operations and, thus, about the functional properties of the pathways connecting different regions of brain.

Recordings obtained from electrodes on the surface of the head contain the simultaneous activities of large numbers of neurons distributed in a large volume of tissue. To separate and localize signals arising from neurons in different parts of this tissue, one must record those signals from a number of different places. One can then attempt to infer the locations of sources from maps of the surface activity.

The importance of considering the locations of voltage sources in event-related potential (ERP) research has become increasingly evident. For example, the familiar P3 ERP of Sutton et al. (6) is now thought to comprise a family of several long-latency positivities, at least some of which may be distinguishable by their surface distributions. These waves overlap one another in time and do not behave similarly. Indeed, P3b (which may be the P3 usually studied) and the tonic Slow Wave bear opposite relationships to behavior: P3b is large when a task is easy and decisions are confident and small when a task is difficult and decisions are unsure. Slow Wave, on the other hand, is small when the task is easy and decisions are confident and large when the task is difficult and decisions are unsure (see references 7-9 for reviews).

Because several, different P3 generators can be active in the same experiment, no unambiguous way exists to interpret changes in measurements obtained from a single recording site. An increase in task difficulty, for example, may increase the amplitude of a globally measured P3 response, decrease it, or leave it unchanged, depending on the relative contribution of each active generator. Similar considerations may apply to the interpretation of auditory N1 responses, which Naatanan and Picton (10) have suggested comprise the activities of as many as five different intracranial voltage generators.

1. RECORDING CONSIDERATIONS

This section outlines several aspects of the electronecephalogram (EEG) recording process that become important in mapping studies. The topics discussed here are (a) the number of recording electrodes, (b) the choice of the voltage reference, and (c) the surface field distortions introduced by electrical artifact filtering systems.

RECORDING ELECTRODES

How many electrodes must be used in order to avoid undersampling the shape of a surface field? This question is important, because undersampling an image does not simply cause information to be lost; it creates artifactual textural features by aliasing the higher spatial frequencies of the original data (their fine-texture structure).

As yet, this question has no definitive answer. Aliasing will not occur if one observes Shannon's rule and samples at two or more times the highest frequency in the data. Unfortunately, the spatial frequency composition of the EEG is essentially unknown, and the opinions of various investigators differ by nearly an order of magnitude. Thus, Desmedt and Bourguet (3) suggest that 17 properly located electrodes will suffice for somatosensory responses. Srebro (11), however, recommends spacing electrodes about 2.5 cm apart when recording visual ERPs, and Gevins and Doyle (cited in reference 12) argue that the critical distance is actually nearer to 2 cm. The latter figure implies an array of about 120 electrodes.

REFERENCE ELECTRODES

The choice of the reference electrode is important in topographic studies because differential EEG amplification effectively subtracts the reference electrode's potential (relative to Earth) from that of each recording electrode. Thus, the surface fields actually recorded will vary according to the nature of the activity present at the reference electrode. This is not necessarily a problem in experiments whose purpose is to test models of underlying voltage sources. In such cases, reference potentials, if known, can be taken into account when models are fit to the data. An active reference does, however, pose a problem in the visual interpretation of surface maps, as it burdens the observer with mentally transforming a two-dimensional surface (or coloration pattern) before interpreting the graphic display.

When the reference site is lateralized relative to the midline, as is the familiar earlobe reference, fields can be distorted in ways that may be difficult to distinguish from true hemispheric asymmetries. The commonly used method of referring recordings to linked earlobes or mastoids does not solve this problem because connecting these electrodes shunts current between the two sides of the head, which also is likely to distort surface fields (13). Digitally linking bilateral earlobe references may avoid this problem (A.S. Gevins, EEG Systems, Inc., San Francisco, CA, personal communication, August, 1986).

Nose-reference electrodes, although commonly thought to be fairly inactive, probably record some portion of the same volume-conducted activity recorded by frontally placed electrodes. This may be expected to reduce the magnitudes of responses obtained from frontally placed recording electrodes. A nose reference might also increase the relative magnitudes of visual responses recorded from occipital leads by picking up activity from the underside of a longitudinally oriented dipole in the occipital pole. The commonly used earlobe and mastoid references can be expected to bias recordings analogously, reducing the amplitudes of responses from dipoles with axes that run generally perpendicularly with respect to the lateral surface of the head. Earlobe and mastoid references may also invert the polarities of responses originating on the superior surface of the temporal lobe (e.g., 14,15). This is presumably due to their relatively ventral locations and proximities to the temporal lobe.

Balanced noncephalic references (16,17) represent possible alternatives. These require nulling the large electrocardiogram (EKG) artifact that occurs when the reference is placed a distance from the scalp. This is usually done using a variably resistive "balancing" circuit, but scalp electrodes placed a distance from one another may require different settings of the nulling circuit. A solution to this problem in multiple-electrode recordings might be to record the EEG along with the EKG artifact, estimate the magnitude of the EKG artifact contaminating the EEG at each electrode, and then subtract an appropriately scaled portion of the recorded EKG from the output of each EEG electrode (e.g., 17).

This approach has limitations, including computational expense and possibly a large increase in noise. The latter derives from the fact that the EKG artifact that contaminates the EEG when a non-scalp reference is used is very large relative to the magnitude of the EEG. This means that the gains of EEG amplifiers must be reduced to avoid distorting the EKG artifact—if the EKG artifact is distorted, the scaling constants needed to compensate the EEG cannot be estimated. Doing this, however, will increase the background noise of the recording by an amount depending on the gain-independent noise levels of the amplifers and the noise introduced by the post-amplification circuitry. Quantization noise associated with the analog-to-digital conversion of the EEG will also increase.

Common-average or "reference-free" recording has been suggested as a solution to the problem of the reference electrode (4,18). Common-average recording involves recording in bipolar fashion from a number of electrodes, all referred to a single site. One then calculates the grand mean EEG waveform, by averaging across electrodes, and subtracts the result pointwise from the EEG recorded at each electrode. Activity recorded by the reference electrode is theoretically of equal magnitude in the mean- and individual-electrode waveforms. Consequently, the effect of the reference electrode should be eliminated from each recording electrode's output when the common-average waveform is subtracted.

A problem with common-average referencing is the difficulty of interpreting changes in field distributions across time. Because voltage generators in different parts of the brain are active at different times, the average voltage recorded from a number of electrodes changes with time. (The common average would be constant if one could record the positive and negative portions of each sequential field in an ERP equally, which would cause the common average to be zero.) Consequently, one cannot know whether a change in amplitude observed at one common-average-referred electrode is due to a change in the (Earth referred) potential of that electrode or whether it is due to a combination of effects at other electrodes. This uncertainty reduces the anatomical value of topographic displays derived from common-average-referred recordings. Moreover, the reference potential is determined by the specific combination of recording sites employed in a particular study. Hence, comparing the data from two studies can be impossible unless the recordings were made with identical electrode arrays.

Hjorth (19) has noted that calculating the Laplacian transform of the recording electrodes' output, or an approximation thereof, may provide a partial answer to the reference problem. Hjorth's approximation amounts to subtracting, from each electrode's output, a distance-weighted portion of the voltage recorded from each of the other electrodes in an array. One feature of this technique is its ability to suppress potentials from outside the immediate vicinity of the recording electrode. Moreover, if the sum of the weights applied to the voltages subtracted equals the weight applied to the site under consideration, the activity at the reference electrode will be removed by the subtraction, and the recording will be reference-free.

DISTORTION CAUSED BY OCULAR ARTIFACT FILTERS

Standard practice in ERP studies includes monitoring eye movements by recording at least one channel of electro-oculogram (EOG) along with the EEG. The EOG is used either as a criterion for excluding EEG data that may contain electrical artifacts or as data in attempts to compensate the EEG for the effects of eye movements (as described, e.g., in reference 20).

Recording the EOG from vertically and horizontally arranged pairs of electrodes allows one to compensate separately for the effects of the vertical and horizontal components of eye movements. This is of some importance in mapping. For example, horizontal eye movements can produce bilaterally asymmetrical waves in the EEG which, if systematic, might be interpreted as functional, hemispheric differences.

The locations of the EOG reference electrodes merit consideration in any artifact reduction scheme based on EOG recordings. Because these procedures involve subtracting a portion of the EOG from the EEG, it is important to minimize the amount of EEG recorded by the EOG electrodes. If the EOG is referred to a distant recording site, the EOG recording will contain frontal EEG as well as EOG. Thus, subtracting the recorded EOG from the EEG during artifact compensation will cause a portion of frontal EEG to be subtracted from the output of each EEG electrode. Because ocular artifacts are large frontally, comparatively large amounts of EOG must be subtracted from the frontal EEG electrodes to compensate for ocular artifacts. As a result, the process of ocular artifact compensation may substantially reduce the amplitudes of responses recorded from frontal electrodes.

A partial solution to this problem is to record the EOG differentially from pairs of electrodes located near enough to one another that the local EEG is essentially identical in each. Differential amplification will then tend to remove the common EEG from the EOG recording. Placing the EOG electrodes near the midline may also help minimize the possibility of introducing artifactual hemispheric asymmetries.

A second possibility is to forego ocular artifact compensation and simply discard epochs in which ocular artifacts are found. This approach avoids some of the assumptions of ocular artifact compensation. This is costly of data. Moreover, it requires the questionable assumption that one can reliably detect small eye movements that might seriously bias ERP measurements were they to occur time-locked to stimulus onsets.

2. MECHANICS OF MAPPING

This section outlines some techniques that can be used to implement an EEG mapping system. A number of potential problems and sources of artifact are discussed. The first subsection describes a general procedure that can be used to construct and display a map of the scalp surface. The second subsection discusses techniques for estimating voltages at locations that are not electrode sites. The third subsection discusses some methods that have been used to measure distances between points on map surfaces.

MAPPING VOLTAGE LEVELS

Despite the expense of commercial EEG topography units, a computer program for EEG mapping is remarkably easy to write. A small microcomputer with reasonable graphics capability is sufficient. The description here assumes that the computer to be used has bit-mapped graphics capability. This is not absolutely necessary, however, as characters can be used to denote voltage levels, with visually darker characters denoting more extreme voltages.

In this discussion, we will assume that the plan of the display calls for representing different voltages with different colors. A color spectrum is frequently used to code voltages. For example, extreme positive voltages can be denoted by reds, near-zero voltages by yellows, and extreme negative voltages by violets. An alternative scheme, which may be easier to interpret, is to code positive voltages with one color and negative voltages with another color, indicating magnitudes by saturation or brightness. A third technique, which is useful for monochrome displays and dot-matrix printers, is to use dot densities to represent voltage levels (reference 1 outlines a procedure of this type).

The first step is to obtain coordinates for drawing the two-dimensional head picture with the sites of the electrodes properly placed in the coordinate system. An accurate International 10/20 System electrode chart is a possible source for a set of head coordinates (this approach is used in reference 3, for example). A technique for obtaining a set of "average" head measurements from real heads, which is probably a superior method, is described by Buchsbaum et al. (1).

That paper also contains charts of the relative positions of electrodes mapped onto the brain. A more accurate but demanding approach is to use an x,y,z positioner to digitize the locations of electrodes for each subject individually. One can then calculate their distances and project the results onto a two-dimensional surface for purposes of mapping. This latter procedure should be considered where time and resources permit.

Assuming that one will use a single, two-dimensional map for all subjects (e.g., the top-down 10/20 map), the next step is to get the map into the computer. An easy way to do this is as follows:

1. Draw a vertical line between the highest point (the largest y value) and the lowest point (the smallest y value) on the map.

2. Starting at the top of the map where the vertical line intersects the outline of the head and working from top to bottom (from y-high to y-low), get the x coordinates for the left and right edges of the head outline at a number of equally spaced values of y. The more x-value pairs obtained, the smoother will be the outline of the head. Sixty-four pairs produce an outline that does not look too bad. This step can be done automatically and with arbitrary precision if the head map is circular, like the 10/20 top view. One might interpolate between the x values, linearly or perhaps with splines, to increase the resolution of an irregularly shaped outline. Bit-mapped imaging is made easier by using a number of x-value pairs equal to the ultimate height of the map in screen pixels. If the map will be drawn with characters, use a value equal to the screen height of the map in units of character height (text rows).

3. After entering the x values of the left and right sides of the map, scale them to fit the x-coordinate system of the display device. The smallest x-low should be the screen coordinate of the leftmost point on the map outline; the largest x-high should be the rightmost coordinate. Add a constant to y-low and y-high so that their values correspond to the lowest and highest rows of pixels in the screen-displayed map. The values between each pair of x values will now correspond to a set of pixel locations comprising a horizontal strip in the screen-displayed map. There are yhigh-ylow+1 such strips. The map is built by stacking the set of strips atop one another, from y-low to y-high.

4. Enter the x and y coordinates of the electrodes into the computer and scale them to the screen coordinate system as was done for the coordinates of the head outline.

5. To display the map, turn on each screen pixel in the stack of strips, one after another. A Fortran 77 program that does this is very simple:

```
          program mapdrawer

*  This program will draw a brain map inside the outline
*  given by the coordinate system.

integer x, ylow, yhigh, xlow(64), xhigh(64)
integer abscissa, ordinate, pixelcolor, yoffset
          .
          .
          .
*  Normally we would adjust the pixel color during each pass
*  through the following do-loop, to make it represent the voltage
*  of its corresponding map point. For now, we will just set the
*  color of each pixel to whatever color corresponds to a color
*  code of 1.
      yoffset = ylow - 1
      do 100 ordinate = ylow, yhigh
         do 50 abscissa = xlow(ordinate), xhigh(ordinate)
            pixelcolor = 1
```

```
          call pixelroutine (abscissa, ordinate, pixelcolor)
50      continue
100   continue
            .
            .
            .
      stop
      end

      subroutine pixelroutine (abscissa, ordinate, pixelcolor)

*  This subroutine turns on one screen pixel at screen
*  coordinates (abscissa, ordinate), and gives it the color
*  corresponding to the color code 'pixelcolor'.  The routine's
*  code depends on the computer and display device used.
            .
            .
            .
      return
      end
```

INTERPOLATING VOLTAGES AT LOCATIONS BETWEEN ELECTRODES

The number of electrodes used in recording determines the number of actual voltage measurements used in constructing a map. This number will usually be many times smaller than the number of picture elements (pixels or characters) in the map. Thus, one must interpolate the voltages between electrode sites to estimate the field strengths at points where no direct measurement is available.

Two general types of interpolation are linear and nonlinear. Each has advantages and disadvantages. Linear interpolation is simpler and computationally less expensive. Its results are also easier to predict, which can simplify identifying display system artifacts. Nonlinear interpolation (described well in reference 5) has an important advantage in that it can assign voltages to points that exceed the voltages at surrounding points. Thus, nonlinear techniques can correctly locate voltage maxima that occur between electrode locations. Linear schemes based on distance-weighted voltage averaging cannot do this because the distance-weighted average of a set of voltages must be less than the maximum voltage in the set. Hence, the maximum of a focus of activity will always be localized at an actual recording site. (Nonlinear techniques, however, can also mislocalize voltage maxima.)

Most, if not all, of the techniques currently used in EEG mapping base their voltage interpolations on measurements from electrodes near the site whose voltage is being estimated. The usual procedure is to use only the voltages obtained from the 3, 4, 5, or n electrodes nearest the map location under consideration (n-nearest-neighbor procedures). This results in important computational savings and guarantees that voltages at very distant locations do not enter into the estimates (although the usual procedure of distance weighting accomplishes much of the latter anyway). A four-nearest-neighbor version of this algorithm is employed in program "map," the source listing of which is included in the Appendix.

A consideration in nearest-neighbor interpolating concerns whether or not the electrodes included in a set of nearest neighbors are required to be located at substantially different angles relative to the location at which a voltage is being estimated. Strict nearest-neighbor interpolation considers only distance in the definition of the set of contributing electrodes. If the recording electrodes are not evenly spaced across the map, the n electrodes nearest some points are likely to cluster in small angular ranges. This can cause abrupt transitions in the estimated voltages at boundaries between regions with different sets of nearest electrodes.

For example, suppose that the n electrodes nearest a point are clustered to one side of the point under consideration. The resulting interpolation is likely to be a poor estimate of the point's actual voltage. Moreover, as the process of estimation moves from left to right, a point will be encountered where a new electrode suddenly joins the set of nearest neighbors and one of the old electrodes drops out. Should the new electrode be located a substantial distance from the set of old neighbors, an artifactual discontinuity in the sequence of interpolated values may result. This artifact appears in the form of arc- or bite-shaped edges at boundaries between sets of neighbors.

A way to reduce this problem is to select the n neighbors so that only one electrode is chosen from each 360/n-degree region surrounding the point to be interpolated (e.g., one from each quadrant in four-nearest neighbor interpolating). This adds complexity to the algorithm and time to the computations but is likely to improve the resulting maps.

A further source of artifact introduced by interpolating occurs at the edges of maps where the locations of the nearest electrodes necessarily cluster toward the center of the map. A direct approach to dealing with this source of artifact is to not plot points that lie beyond the outermost electrodes.

OBTAINING POINT-TO-ELECTRODE DISTANCES

Several methods of obtaining distances seem to be in use. The different procedures differ in complexity and accuracy. Evidently, most of the techniques in use proceed by projecting the voltages of electrodes onto a standardized map of the head and then computing distances on the flat surface of the resulting figure.

One approach is to project a representation of a spherical head with the 10/20 system electrode locations represented on it onto a two-dimensional surface. Then one can use the Pythagorean Theorem to find the interelectrode distances (e.g., reference 3). This approach is straightforward and useful, despite its anatomically schematic nature. It does, however, tend to distort distance measurements taken between points near the periphery of the map.

Another method has been described by Buchsbaum et al. (1). These investigators measured actual heads, divided them into imaginary coronal sections 1-cm apart, and measured distances along the strips of scalp thus defined to locate the positions of electrodes. The head surface was projected onto two dimensions by peeling the imaginary "strips" of scalp thus defined and laying them flat. As the authors note, this technique somewhat distorts the view of the frontal and occipital poles. It should, however, yield better approximations along the dorsal-ventral axis than those obtained from a spherical projection.

3. USING THE EVOKED RESPONSE PROGRAMS

PROGRAM OUTLINE

The ANSI-Standard Fortran 77 programs "neighbors" and "map," included in the Appendix, perform the operations necessary to compute a simple four-nearest neighbor map by linear interpolation. These routines can also be used to map transformations of the data, such as the results of spectrum analyses, spatial factor analyses, and so on.

The programs and the calculations they perform are quite simple. Difficulties encountered in understanding them can be reasonably attributed to the machine-specific properties of the graphics routines and to the author's programming style.

The programs were written for a MASSCOMP 5500 computer with color graphics terminal. Converting the programs to another machine will require replacing the graphics routines in the program "map." The standardization of Fortran 77, unfortunately, has not been completely successful; thus some non-graphics sections of the programs may need rewriting before they will compile on another machine. As most of the programs' content is fairly elementary Fortran, however, compiler incompatibilities may be limited to input and output statements.

The program "map" uses the geometric coordinates of the recording electrodes, which are stored in a data file called "trodefile," to plot the EEG voltages stored in file "eeg" onto the map. The coordinates of the map's left and right sides are stored in files "head.l" and "head.r," respectively. The interpolation of distances is carried out in the two-dimensional plane defined by the map outline.

The resulting distances, one from each plotted point in the map to each of its four nearest neighbors, are stored in the data file named "neighbordata." This file is the output of program "neighbors." The map coordinates of the recording electrodes are stored in the file named "trodefile," which the user must supply, and which is used both by "neighbors" and "map." The interpolating procedure used by program "map" is that of Buchsbaum et al. (1, pp. 238-239), who describe its operation in detail and provide a computational example (see also Section 2).

The file named "eeg" is free formatted for easy examination and editing. Implementing the program with "eeg" formatted this way will simplify debugging. To use the program, however, one must first copy a set of voltages into "eeg" from a primary data file. The data in "eeg" should comprise the voltages from m electrodes obtained at a single point in time.

They are written into "eeg" according to electrode number, with the first eeg data point corresponding to the first electrode whose coordinates are listed in "trodefile," and so forth.

After debugging, substantial savings in running time can be achieved by rewriting the file-reading routine to read directly from an unformatted ("binary") data file containing the original, evoked-potential voltage time series. A second savings can be achieved by causing the program to store the point-to-electrode distances in memory, loop back after graphing, and ask the user for another set of voltages to map. This latter approach will save disc access time whenever more than one map is to be displayed.

The data files "head.1" and "head.r" contain the x coordinates of the left and right sides of the map outline, respectively. The first entry in each file is the number of x coordinates in the file. The second entry is the lowest x coordinate of the left (in head.1) or right (head.r) side of the map. Subsequent entries in each file are the second, third, ... , and m-th x coordinates of the left and right map-outlines. These files are used by both "neighbors" and "map."

RUNNING THE PROGRAMS

Prior to running, the coordinates of the map and electrodes must be entered into the files "head.1," "head.r," and "trodefile," as described in the previous section. One then runs the compiled version of "neighbors," which generates the data file of point-to-electrode distances. Program "neighbors" must be rerun any time the head map outline is modified and anytime the set of electrode locations in "trodefile" is changed.

One then runs the mapping program. Program "map" reads "eeg," "head.1," "head.r," "trodefile," and "neighbordat." It then interpolates the voltage at each point on the map, assigns a color to that voltage, assigns that color to a 4 x 4 square of pixels at the corresponding set of screen coordinates, and exits.

REFERENCES

1. Buchsbaum, M.S., Rigal, F., Coppola, R , Cappelletti, J., King, C., and Johnson, J., "A New System for Gray-Level Surface Distribution Maps of Electrical Activity." Electroencephalography and Clinical Neurophysiology, Vol. 53, pp. 237-242, 1982.

2. Coppola, R., Buchsbaum, M.S., and Rigal, F., "Computer Generation of Surface Distribution Maps of Measures of Brain Activity." Computers in Biology and Medicine, Vol. 12, pp. 191-199, 1982.

3. Desmedt, J.E. and Bourguet, M., "Color Imaging of Parietal and Frontal Somatosensory Potential Fields Evoked by Stimulation of Median or Posterior Tibial Nerve in Man." Electroencephalography and Clinical Neurophysiology, Vol. 62, pp. 1-17, 1985.

4. Lehmann, E. and Skrandies, W., "Spatial Analysis of Evoked Potentials in Man - A Review." Progress in Neurobiology, Vol. 23, pp. 227-250, 1984.

5. Perrin, F., Pernier, J., Bertrand, M.H., and Echallier, J. F., "Mapping of Scalp Potentials by Surface Spline Interpolation." Electroencephalography and Clinical Neurophysiology, Vol. 66, pp. 75-81, 1987.

6. Sutton, S., Braren, M., Zubin, J., and John, E.R., "Evoked Potential Correlates of Stimulus Uncertainty." Science, Vol. 150, pp. 1187-1188, 1965.

7. Ruchkin, D.S., Sutton, S., and Mahaffey, D., "Functional Differences Between Members of the P300 Complex: P3e and P3b." Psychophysiology, Vol. 24, pp. 87-103, 1987.

8. Sutton, S. and Ruchkin, D.S., "The Late Positive Complex: Advances and New Problems." In R. Karrer, J. Cohen, and P. Tueting (Eds.), Brain and Information: Event-Related Potentials, Annals of the New York Academy of Sciences, Vol. 425, pp. 1-23, 1984.

9. Ruchkin, D.S. and Sutton, S., "Positive Slow Wave and P300: Association and Disassociation." In A.K.W. Gaillard and W. Ritter (Eds.), Tutorials in ERP Research: Endogenous Components, North Holland, Amsterdam, 1983, pp. 233-250.

10. Naatanen, R. and Picton, T., "The N1 Wave of the Human Electric and Magnetic Response to Sound: A Review and an Analysis of the Component Structure." Psychophysiology, Vol. 24, pp. 375-425, 1987.

11. Srebro, R., "Localization of Visually Evoked Cortical Activity in Humans." Journal of Physiology (London), Vol. 360, pp. 233-246, 1985.

12. Gevins, A.S. and Cutillo, B.A., "Signals of Cognition." In F.H. Lopes da Silva, W. Storm van Leewuen, and A. Remond (Eds.), Clinical Applications of Computer Analysis of EEG and other Neurophysiological Signals, Elsevier Press, New York, NY, 1986, pp. 335-384.

13. Nunez, P.L., *Electric Fields in the Brain: The Neurophysics of EEG,* Oxford University Press, New York, NY, 1981.

14. Vaughan, H.G. and Ritter, W., "The Sources of Auditory Evoked Responses Recorded from the Human Scalp." *Electroencephalography and Clinical Neurophysiology,* Vol. 28, pp. 360-367, 1970.

15. Wood, C.C. and Wolpaw, J.R., "Scalp Distribution of Human Auditory Evoked Potentials. II. Evidence for Overlapping Sources and Involvement of Auditory Cortex." *Electroencephalography and Clinical Neurophysiology,* Vol. 54, pp. 25-38, 1982.

16. Stephenson, W.A. and Gibbs, F.A., "A Balanced Non-Cephalic Reference Electrode." *Electroencephalography and Clinical Neurophysiology,* Vol. 3, pp. 237-240, 1951.

17. Nakamura, M. and Shibasake, H., "Elimination of EKG Artifacts from EEG Records: A New Method of Non-Cephalic Referential EEG Recording." *Electroencephalography and Clinical Neurophysiology,* Vol. 66, pp. 89-92, 1987.

18. Offner, F.F., "The EEG as Potential Mapping: The Value of the Average Monopolar Reference." *Electroencephalography and Clinical Neurophysiology,* Vol. 2, pp. 215-216, 1950.

19. Hjorth, B. "An Online Transformation of EEG Scalp Potentials into Orthogonal Source Derivations." *Electroencephalography and Clinical Neurophysiology,* Vol. 39, pp. 526-530, 1975.

20. Gratton, G., Coles, G.H., and Donchin, E., "A New Method for Off-Line Removal of Ocular Artifact." *Electroencephalography and Clinical Neurophysiology,* Vol. 55, pp. 468-484, 1983.

APPENDIX

```
      program neighbors

* This program finds the coordinates of the four recording sites
* nearest each point on the surface of a map. It then finds the
* distance from each point to the four nearest recording sites.

* The coordinates of the set of map points are the
* coordinates of the points between the left and right edge
* sides of the map.  The values of these coordinates are
* read from the files, head.l and head.r.

* To run the program type:  neighbors <RETURN>

* The program will then ask for the number of EEG channels
* to be used in the analysis, and the name to be given to
* the data file that will contain the results.

* Inputs:

* nc = The no. of eeg channels.  Obtained from the keyboard.
* neighbordata = The name of the file to which the program's
*   output is written.  Obtained from the keyboard.
* trodex(c) = The real-valued abscissa of the c-th electrode.
*   Obtained from file 'trodefile'.
* trodey(c) = The real-valued ordinate of the c-th electrode.
*   Obtained from file 'trodefile'.
* xlow(i) = The integer-valued coordinate of the left edge of
*   the map outline at the i-th value of the map ordinate.
*   Obtained from file 'head.l'.
* xhigh(i) = The integer-valued coordinate of the right edge of
*   the map outline at the i-th value of the map ordinate.
*   Obtained from file 'head.r'.
* nyl = The number of xlow-xhigh pairs in the two headfiles.
*   Obtained from file 'head.l'.
* nyr = The number of xlow-xhigh pairs in the two headfiles.
*   Obtained from file 'head.r'.

* Outputs:  Are written to the file whose name is contained
*  in the variable 'neighbordata'.  This file is direct access
*  and contains one record per point in the headmap.  The records
*  are ordered from xlow to xhigh, and from ylow to yhigh.  (The
*  abscissa values vary most rapidly.)  The data in each record
*  are:

* x = The abscissa of a map point.
* y = The point's ordinate.
* neighbor(i) = Four values are written.  The i-th value is
*   the channel number of the point's i-th nearest neighbor.
* d(i) = Four values are written.  The i-th value is the
*   distance to the point's i-th nearest neighbor.
* siteflag = 1 if the point is an electrode site, 0 otherwise.
```

```
      integer siteflag,ylow,yhigh
      integer x,y,xlow(200),xhigh(200)
      integer c,nc,jn,neighbor(4)
      integer jflag,nn,pointno
      real trodex(32),trodey(32)
      real dsq(32), d(4), ndsq(4)
      character*80 neighbordata

* Input EEG and output data

      print *, 'No. of EEG channels?'
      read *, nc
      print *, 'Name of file to write neighbor data to?'
      read *, neighbordata

* Read files containing head data

      open (3,file="head.l")
      open (4,file="head.r")
      rewind (3)
      rewind (4)

      read (3,*) nyl
      read (4,*) nyr

      if (nyl .ne. nyr) then
        print *, "Error: head files unequal."
        stop
      end if

      do 100 i = 1, nyl
        read (3,*) xlow(i)
        read (4,*) xhigh(i)
100   continue

      close (3)
      close (4)

* Here we read a file of electrode coordinates

      open (3, file="trodefile")
      rewind (3)

      do 200 c = 1, nc
        read (3,*) trodex(c), trodey(c)
200   continue

      close (3)

* Find the 4 nearest neighbor electrodes for each point
* in the head map and write data to file

      ylow = 1
      yhigh = nyl
      jn = 4
```

```
* Open the data file
        jrecl = 52
        open(3,file=neighbordata,status='new',access='direct',
     +    form='unformatted',recl=jrecl)
        rewind (3)

        pointno = 0

        do 600 y = ylow, yhigh

          do 500 x = xlow(y), xhigh(y)
            pointno = pointno + 1
            siteflag = 0

* Calculate distance from the current x-y to each trode
            do 300 c = 1, nc
              dsq(c) = (trodex(c)-float(x))**2.
     +        + (trodey(c)-float(y))**2.
              if(dsq(c).eq.0.) siteflag = 1
300         continue

* Find the first neighbor
            neighbor(1) = 1
            ndsq(1) = dsq(1)
            do 350 c = 1, nc
              if(dsq(c).lt. dsq(neighbor(1))) then
                neighbor(1) = c
                ndsq(1) = dsq(c)
              end if
350         continue

* Find subsequent neighbors
            do 450 nn = 2, 4

* Find a candidate for next nearest
              neighbor(nn) = 1
405           nflag = 0

              do 410 jtempnn = 1, nn-1
                if (neighbor(nn) .eq. neighbor(jtempnn)) then
                  nflag = 1
                end if
410           continue

              if (nflag .eq. 1)  then
                neighbor(nn) = neighbor(nn) + 1
                go to 405
              end if

* Check the channel coordinates for nearer neighbors
              jsomebetter = 0

              do 430 c = 1, nc

* See if c is a better candidate
```

```
                    if(dsq(c).lt.dsq(neighbor(nn))) then

* If so, flag it if it's already a nearer neighbor
                      jflag = 0

                      do 420 jtempnn= 1, nn-1
                        if(c .eq. neighbor(jtempnn))then
                        jflag = 1
* If we're here we've decided the current candidate
* really isn't better
                        end if
420                   continue

* If not, call it the next nearest neighbor
                      if(jflag.eq.0) then
                        jsomebetter = 1
                        neighbor(nn) = c
                        ndsq(nn) = dsq(c)
                      end if

                    end if

430               continue

                  if (jsomebetter.eq.0) then
                    ndsq(nn) = dsq(neighbor(nn))
                  end if

                  jsomebetter = 0

450             continue

* Convert squared distances to distances
                do 480 nn = 1, 4
                  d(nn) = sqrt(ndsq(nn))
480             continue

* Write the data to file
                write(3,rec=pointno)x,y,neighbor,d,siteflag

500           continue

600       continue

          close (3)

          stop
          end
```

```
      program map

* This program maps the voltages read from a set of EEG
* electrodes, interpolates voltages at locations on a map
* between electrodes, assigns a color code to the voltage
* at each point, and graphs the resulting map.

*  To run the program, type:  map <RETURN>

      character*80 neighbordata, eegfile
      integer neighbors(4), siteflag
      integer xx, yy, colorindex, xoff, yoff
      integer c, nc, pointno, colors(64)
      integer x,y,xlow(200),xhigh(200)
      integer headxpairs,ylow,yhigh
      real v(32), d(4)

*  Definitions:

*  headxpairs = The number of map ordinate pairs defining the
*               left and right sides of the head outline.
*  nneighbors = The number of neighboring electrodes
*               used to interpolate voltages.
*  jrecl = The record length of the neighbor data file.
*  xoff = Graphics offset of the head map's abscissa values.
*  yoff = Graphics offset of the head map's ordinate values.
*  eegfile = The name of the file containing eeg voltages.
*  c = An EEG channel number index.
*  nc = The number of channels of eeg to plot.
*  ylow = Index of the first map ordinate value.
*  yhigh = Number of xlow and xhigh pairs.
*        = The number of points on the y axis where x values
*          of xlow and xhigh are defined.
*  xlow(i) = Abcissa of the i-th point on the left edge of the map.
*  xhigh(i) = As above for the right edge.
*  v(c) = The voltage read in channel c.

* Inputs:

* yhigh - Read from head.l.
* iycheck - Read from head.r.  Should equal yhigh.
* xlow(i) - Read from head.l.
* xhigh(i) - Read from head.r.
* v(c) - Read from file eeg.
* xx - Read from neighbordata.  Not used.
* yy - Read from neighbordata.  Not used.
* neighbors(i) - Channel no. of a point's i-th nearest neighbor.
* d(i) - Distance to a point's i-th nearest neighbor.  Read from
*        neighbordata.
* siteflag - Read from neighbordata.  Not used.

* Outputs:  Written to graphics device.
```

```
* Variable initialization.

      ylow = 1
      headxpairs = 65
      nneighbors = 4
      neighbordata = 'neighbordat'
      jrecl = 52
      xoff = 225
      yoff = 166
      eegfile = 'eeg'
      nc = 16

* Read head files.
      open(3,file='head.l')
      open(4,file='head.r')
      rewind (3)
      rewind (4)
      read (3,*) yhigh
      read (4,*) iycheck

      if (yhigh .ne. iycheck) then
        print *, 'Error: headfiles do not match in stated sizes.'
        stop
      end if

      do 500 i = ylow, yhigh
        read (3,*) xlow(i)
        read (4,*) xhigh(i)
500   continue

      close(3)

* Read eeg channel data.  These are formatted for easy editing.
      open (10,file=eegfile)
      rewind (10)

      do 600 c = 1, nc
        read(10,*) v(c)
600   continue

      close (10)

* Here we autoscale the eeg data so that the min and max
* voltages in the file are assigned the most extreme colors
* in the color code spectrum.
      vmin = v(1)
      vmax = v(1)

      do 700 c = 2, nc
        if(v(c) .lt. vmin) vmin = v(c)
        if(v(c) .gt. vmax) vmax = v(c)
700   continue

      vrange = vmax - vmin
```

```
* Scale the voltages to fall between 1. and 24.
        do 720 c = 1, nc
          v(c) = ((v(c)-vmin)/vrange) * 23. + 1.
720     continue

* Open a file of neighbor electrode info for reading.
        open(4,file=neighbordata,status='unknown',access='direct',
     +  form='unformatted',recl=jrecl)
        rewind (4)

* Assign MASCOMP graphics, clear planes.
        call mgiasngp (0, 0)
        call mgiclearpln (0, -1, 0)
        call mgigetvcoor (2, xleft, ybottom, xright, ytop, placed)

* Here we modify the MASSCOMP color map using the Color Naming System
* Utility.  The map is now approximately a color spectrum, with reds
* assigned to low numbers and purples assigned to high numbers.
        colors(1) = mgfcns("very-dark-red")
        colors(2) = mgfcns("darker-red")
        colors(3) = mgfcns("slightly-orangish-dark-red")
        colors(4) = mgfcns("slightly-orangish-red")
        colors(5) = mgfcns("orange-red")
        colors(6) = mgfcns("orange")
        colors(7) = mgfcns("yellowish-orange")
        colors(8) = mgfcns("slightly-yellow-orange")
        colors(9) = mgfcns("yellow-orange")
        colors(10) = mgfcns("orangish-yellow")
        colors(11) = mgfcns("slightly-orangish-yellow")
        colors(12) = mgfcns("yellow")
        colors(13) = mgfcns("greenish-yellow")
        colors(14) = mgfcns("slightly-green-yellow")
        colors(15) = mgfcns("green-yellow")
        colors(16) = mgfcns("yellowish-green")
        colors(17) = mgfcns("slightly-yellowish-green")
        colors(18) = mgfcns("green")
        colors(19) = mgfcns("medium-light-bluish-green")
        colors(20) = mgfcns("greenish-medium-dark-blue")
        colors(21) = mgfcns("greenish-dark-blue")
        colors(22) = mgfcns("darker-dark-blue")
        colors(23) = mgfcns("dark-purplish-dark-blue")
        colors(24) = mgfcns("medium-dark-bluish-dark-purple")
        colors(25) = mgfcns("very-dark-purple")

        call mgicms (1,25,colors)

* Read from the neighbor data file, interpolate voltages, convert
* to colors, and plot the points.
        pointno = 0

        do 1000 y = ylow, yhigh

          jjmult = 4
          do 900 x = xlow(y), xhigh(y)
            pointno = pointno + 1
```

```
            read(4,rec=pointno)xx,yy,neighbors,d,siteflag
            vinterp = 0
            recipsum = 0.
            vint = 0.

            do 850 i = 1, nneighbors
              if (d(i) .eq. 0.) d(i) = 1.
              rd = 1./d(i)
              recipsum = recipsum + rd
              vint = vint + v(neighbors(i)) * rd
850         continue

            vint = vint/recipsum

* Convert the interpolated voltage to color and plot the point.
            colorindex = int(25. - vint+ .5)
            call mgihue(colorindex)
            call mgibox(jjmult*x+xoff,jjmult*y+yoff,
    +       jjmult*x+xoff+jjmult,jjmult*y+yoff+jjmult)
900       continue

1000    continue

* Close the neighbor data file.
        close (4)

* Plot a colorbar
          jy = 26
          do 1050 y = 1, 250, 10
            jy = jy - 1
            call mgihue (jy)
            call mgibox (x,y+180,x+40,y+187)
1050      continue

* This would be a good place to put some orienting
* features on the map, such as a nose or some sulci.

*

* Deassign MASSCOMP graphics
20000   call mgideagp
        call system ('clear')

        stop
        end
```