United States
Environmental Protection
Agency

Environmental Research
Laboratory
Narragansett RI 02882

EPA-600/3-80-010
January 1980

Research and Development

Development of an Interactive System to Study Sub-Lethal Effects of Pollutants on the Behavior of Organisms

RESEARCH REPORTING SERIES

Research reports of the Office of Research and Development, U.S. Environmental Protection Agency, have been grouped into nine series. These nine broad categories were established to facilitate further development and application of environmental technology. Elimination of traditional grouping was consciously planned to foster technology transfer and a maximum interface in related fields. The nine series are:

1. Environmental Health Effects Research
2. Environmental Protection Technology
3. Ecological Research
4. Environmental Monitoring
5. Socioeconomic Environmental Studies
6. Scientific and Technical Assessment Reports (STAR)
7. Interagency Energy-Environment Research and Development
8. "Special" Reports
9. Miscellaneous Reports

This report has been assigned to the ECOLOGICAL RESEARCH series. This series describes research on the effects of pollution on humans, plant and animal species, and materials. Problems are assessed for their long- and short-term influences. Investigations include formation, transport, and pathway studies to determine the fate of pollutants and their effects. This work provides the technical basis for setting standards to minimize undesirable changes in living organisms in the aquatic, terrestrial, and atmospheric environments.

This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

EPA-600/3-80-010
January 1980

DEVELOPMENT OF AN INTERACTIVE SYSTEM TO STUDY SUB-LETHAL EFFECTS OF POLLUTANTS ON THE BEHAVIOR OF ORGANISMS

by

John O. B. Greaves
Department of Electrical Engineering
and
Robert S. Wilson
Department of Biology
Southeastern Massachusetts University
N. Dartmouth, Massachusetts 02747

Grant No. R803244

Project Officer

Don C. Miller
Environmental Research Laboratory
Narragansett, Rhode Island 02882

ENVIRONMENTAL RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
NARRAGANSETT, RHODE ISLAND 02882

DISCLAIMER

This report has been reviewed by the Environmental Research Laboratory, Narragansett, U.S. Environmental Protection Agency, and approved for publication. Approval does not signify that the contents necessarily reflect the views and policies of the U.S. Environmental Protection Agency, nor does mention of trade names or commercial products constitute endorsement or recommendation for use.

FOREWORD

The Environmental Research Laboratory of the U.S. Environmental Protection Agency is located on the shore of Narragansett Bay, Rhode Island. In order to assure the protection of marine resources, the laboratory is charged with providing a scientifically sound basis for Agency decisions on the environmental safety of various uses of marine systems. To a great extent, this requires research on the tolerance of marine organisms and their life stages as well as of ecosystems to many forms of pollution stress. In addition, a knowledge of pollutant transport and fate is needed.

This report describes a three-year development project culminating in the design and full implementation of a computerized data acquisition and analysis system for the study of animal behavior. Movements of organisms are recorded on video tape and later played back into a television computer interface. The user, with full interactive graphics support, extracts quantitative information on the movements of large numbers of organisms. The work cited in this report was essentially complete as of June, 1977. However the system allows new capabilities to be added in the form of software modules, and its repertoire continues to grow. An account of this project has also been presented in an invited paper (10) at the "State of Marine Environmental Research" Symposium, June 1977, EPA Environmental Research Lab in Narragansett, Rhode Island.

ABSTRACT

Experimental investigation of the movements of organisms often entails the acquisition and processing of large samples of spatio-temporal data. An interactive, interpretive, on-line computer television system (viz., the Bugsystem) was initially developed at the University of California at Santa Barbara in order to expedite such analyses. Aspects of the structure of this prototype system are outlined. Its effectiveness is evaluated with regard to the problems confronting the bio-behavioral researcher.

A second generation system has been developed under a research grant from the Environmental Protection Agency. Utilizing new hardware and software, it in many ways constitutes a generalization of its prototype. We describe features of the refined system which provide for the following: a large degree of machine-independence, significant expansion of the size of data records, inclusion of experimental parameters and variables within the data structure, investigation of rotational and flectional movement, statistical analysis, advanced graphics and tracking of organisms in three dimensional space. Current utilization of the Bugsystem for research in behavioral physiology and the potential for future development are discussed.

TABLE OF CONTENTS

FIGURES

ACKNOWLEDGMENTS

The authors would like to thank Dr. William Lang for his friendly assistance in debugging the system and guiding it from its development stages to its fully operational status. Dr. Don C. Miller as Project Officer provided the initiative and support to start and manage the research grant development of the Bugsystem. He superivsed and coordinated the merging of technological skills with the needs of the animal behavior research team and the goals of the Environmental Protection Agency. Dr. Miller and Dr. Lang also provided some of the data shown in the figures of this report. Dr. William Hand offered valuable suggestions during the early development of the system as the first real biological user and contributed a user's manual. The programmers who helped create the extensive software package include: Steve Fyfe, Richard Boucher, Paul Messier, Bob Souza, Peter Roy, Dr. Hans van den Biggelaar, Wallace Shubach, Barbara Blake and David Sleczkowski. Their help and patience was appreciated. Mr. Jay Sinnett was always available for consultation in the electronics area and he provided refinements to the Bugwatcher hardware and built the video multiplexing circuitry for the dual camera data collection scheme. And finally, we would like to thank Dr. Eric Schneider, who as the Director of the Narragansett laboratory, recognized this project as a potentially valuable asset for the Agency and committed himself and the laboratory resources to the successful completion of the project.

SECTION I

INTRODUCTION

The fundamental focus of behavioral research is the description and explication of what individual organisms *do*. Because those biological activities most often classified as "behavior" consist largely (although not exclusively) of the movements of organisms, quantitative investigation of behavior is often dependent upon the collection of temporal sequences of spatial information. Cinematography is the classical method used to gather such information. Behavioral variables such as the position of an organism, its orientation and angles of flection of its appendages are extracted from the motion pictures by means of frame-by-frame analysis. An important advantage of the technique is its flexibility; it may, in principle, be employed in the investigation of any overt behavior which can be photographed. However, two factors preclude the widespread use of this method: (1) manual quantitization via frame-by-frame analysis is a lengthy and tedious process; and (2) once the data are obtained, a substantial amount of subsequent data processing may be required in their analysis.

The Bugsystem (3, 5, 6) was designed to enable the acquisition and processing of video data by a minicomputer. Behavioral data are initially recorded using standard closed-circuit television equipment. These data are analyzed by replaying a video tape into a specially designed video-to-digital processor (christened the "Bugwatcher"). This device acts as an edge detector, greatly reducing the information flow to the computer and thereby allowing the real-time collection of spatial coordinates delineating the outlines of moving organisms. Frame-by-frame analysis of digitized video data is achieved through the use of specially designed programs tailored to the task of quantifying behavioral variables of interest to the researcher.

The original version of the Bugsystem was developed by Greaves and implemented on an IBM 1800 computer at the University of California, Santa Barbara. This prototye system, previously described by Greaves (5, 6), has been utilized by Hand and Schmidt (7) and by Wilson (9) to investigate the photokineses and phototaxes of marine dinoflagellates. However, several features of this system severely limited the domain of its application. Supported by a research grant from the Environmental Protection Agency, we have developed a second generation Bugsystem. Our explicit goal in the design of this system was to provide a *flexible* tool for the quantitative investigation of behavior, a system capable of realizing much of the potential of frame-by-frame analytical techniques.

The purpose of this paper is to describe this second generation Bugsystem, emphasizing the way in which certain hardware and software refinements have expanded the scope of questions which may be conveniently answered by means of "bugwatching". We discuss the way in which the user interacts with the system via a specially formulated "Behavioral Research Language" and the way in which this language has been implemented upon physical machines. We also described the input of data to the system, the processing and display of behavioral data and a variety of experimental strategies accommodated by the system. Finally, we outline work now in progress to provide for the analysis of movement in three dimensions.

DEFINING A BEHAVIORAL RESEARCH LANGUAGE (BRL)

The Behavioral Research Language, or BRL, is a high level operator based language that is tailored to the unique problems associated with the input, scaling, analysis and display of video images of moving objects. BRL is an interpretive language which runs as an application program on a good sized minicomputer and relies heavily upon user interaction with a storage graphics terminal to input, plot, edit and transform the data through image processing functions. Image processing generally culminates in the computation of paths or trajectories of the objects moving before the video camera. Sets of trajectories may be merged together and the data may be transformed to yield time series of behavioral variables (e.g., linear velocity, angular velocity, direction of travel, etc). These results may then be analyzed statistically and the resultant data sets either listed numerically or plotted on the graphics terminal or plotter. Thus, the Bugsystem consists of two basic subsystems: (1) an unique image processing system for the frame-by-frame analysis of video data, and (2) a signal processing system for the statistical analysis of equispaced time series.

The key element to understanding and using the BRL lies in grasping the operator-operand-resultant nature of specifying functions or commands to the system on the graphics terminal keyboard. The general command syntax is as follows:

```
*Operator/sw/sw Operand-name/sw/sw Resultant-name/sw/sw nl, . . ., n5
```

Where "*" is the prompting character, "Operator" is one of the available functions (of which there are currently 96, with the list still growing), "Operand-name" is the name of the input (or operand) data set, "Resultant-name" is the name of the output (or resultant) data set, and the numeric constants "nl" through "n5" are optional numeric constants which govern details of the function of certain operators. The "/sw" denote optional "switches" (the word is taken from computer jargon), which are used as operator modifiers or to supply special information to the operator being used. All data sets are disk resident and are specified by a four-letter name that can be used to denote an experimental condition, a two-letter extention that specifies the type of data represented (e.g., "VI"-video, "PA"-path, "LV"-linear velocity, "CA"-categorized, etc.) and a six-letter front name that can be used to indentify the species studied and/or the date of the experiment. The front name must be specified only when starting the system and remains unchanged unless it is explicitly modified by the LOAD operator. Operator names will henceforth be underlined in the text.

For example, to plot all of the data in the video file called BUGS.VI, the command would appear as "*PLOT BUGS.VI". To plot only the first twenty frames of this file, one would enter "*PLOT BUGS.VI 1, 20". To calculate the linear velocities for the paths in the file BUGS.PA, the command line is "*LVEL BUGS.PA BUGS.LV". Note that in this language the loops required to process all data elements within a data set are not explicitly stated. The operator automatically processes all of the elements of the operand data file unless directed to a particular subset (e.g., "*PLOT 1, 20", as illustrated above).

Three other aspects of BRL are worth including here. The first of these concerns the way in which data are represented within the Bugsystem: a data set consists of one or more vectors of variable length. While performing image processing operations, each vector represents one video frame; one element of such a vector represents a single point in two dimensional space. As the analysis of the data proceeds through successive application of operators to operand data sets, a single vector may represent an organism's path (i.e., a time series of Cartesian coordinates in two space as in the file "BUGS.PA" illustrated above) or a real function defined over the length of such a path (e.g., the estimate of instantaneous linear velocity as previously illustrated by the file "BUGS.LV"). Finally, in the statistical analysis of such data a vector may correspond to a set of statistical parameters, a collection of "bins" or categories established for histograming, an estimate of an autocorrelation function, etc.

The second aspect of the language to be considered here are the self-documenting aspects of BRL. No one can be expected to memorize all of the 96+ operator names, what they do in detail and the various switches and numeric constants which they expect. To help in this regard, the NAMES operator lists on the terminal the names of all the keyboard operators. Moreover, entering "*Operator/HELP" for any of the available operators will cause the system to print a full page of information describing what the operator does, the types of operands for which the operation is defined, what constants and switches are expected and an example of the operator's use.

The third aspect of BRL to be considered involves the construction of user programs. BRL was designed primarily to be an interactive language: the user normally enters commands at the terminal one at a time and thereby directs the analysis to its desired end. It is also possible to create a disk file of commands consisting of the operators and operand specifications and to direct the system to execute this stored sequence of operations (a computer program written in BRL) via the USER operator.

THE IMPLEMENTATION OF BRL

The implementation of BRL by means of the second generation Bugsystem differs significantly (both in hardware and in software) from that of the prototype system. This section outlines the main features of the new system, giving reasons for their importance.

A New Bugwatcher

The Bugwatcher hardware was redesigned to utilize medium scale integration (MSI) circuitry. Algorithmic state machine (ASM) charts were employed to formulate and to document the new design. Functionally, the new Bugwatcher is similar to its earlier counterpart: it extracts digital X-Y coordinate pairs representing points belonging to image outlines from the video raster scan and stuffs these coordinates into the computer's direct memory access (DMA) channel. The computer program to input these data buffers them and writes them to disk, thereby allowing the collection of extremely long data records. Within such a record each frame of video data is represented as a vector. Each video vector sent by the Bugwatcher to the computer contains not only a variable length list of coordinate data, but also includes a leading header of fixed length. The elements of this header are refered to as "vector attributes" and are employed to encode information associated with each vector. Video data possess four 16-bit words of attribute information supplied by the Bugwatcher hardware: (1) a unique word consisting entirely of zeroes used by the software to delimit frame boundaries, (2) a descriptor word which contains an encoding of the frame rate at which the video data were digitized and the on-off status of tone stimulus markers, (3) a total frame counter that can be used to determine relative or absolute time intervals between data segments recorded at varying time intervals, and (4) auxiliary digital input which allows an encoding of an experimental variable (e.g., the direction of the source of stimulation) to be automatically associated with each video frame. Representation of stimulus conditions is discussed in more detail below (see "Coupling to Research Environments"). This division of vectors into attributes and data applies to all vectors manipulated by Bugsystem software. However, the meanings of each attribute depends on the type of data; e.g., one of the attributes of a path (represented as a single vector) is the starting frame number.

The Implementation Language--FORTRAN IV

The prototype BRL system was implemented entirely in assembler language on an IBM 1800 computer and ran as a stand-alone system with no operating system support. This made it non-portable and difficult to maintain and expand. The new BRL system was designed to overcome these major shortcomings. We chose FORTRAN IV as the language with which to implement the Bugsystem. FORTRAN IV has become a standard language among minicomputers and has simplified the tasks of maintaining Bugsystem software and training new programmers to implement new BRL operators. The importance of these aspects cannot be overestimated since, all in all, ten different programmers have added software to the system during its three years of development. Each required instruction on the software conventions and use of system utililty subroutine packages, but at least they knew the FORTRAN language.

The software was first operational on a PDP 11/45 computer at Southeastern Massachusetts University under the DOS-9 operating system. It was then implemented on a Data General ECLIPSE S/200 under the RDOS operating system. Some assembler language subroutines had to be recoded, including the software drivers that handle the direct memory channel to the Bugwatcher. But the transition to the new computer went fairly smoothly. The PDP 11 system was maintained for development purposes after the ECLIPSE was moved to the Narragansett EPA Laboratory. Due to malfunctioning DOS-9 software, the PDP 11 operating system was changed to RT 11--a change that required as much or more development effort than changing computers!

Virtual File Structure

Certainly one of the major disadvantages to using minicomputers for large application software projects is the address space limitation imposed by its small word size. This system was no exception. To overcome this limitation so as to allow both programs and data to fit within allocated memory, both program structures and data structures were designed so that only pieces of either resided in memory at any given time. Manufacturers of minicomputers recognize this problem and provide software support for manipulating overlayed programs; i.e., programs consisting of parts which are swapped in and out of main memory. However, software support was not available for similarly overlaying data sets. The earlier prototype system did not allow data sets to be any larger than the memory buffer on the IBM 1800. This was not acceptable in the newer system. Certain Bugsystem applications require the acquisition and analysis of behavioral records consisting of many frames of video data or the trajectories of hundreds of organisms. The system required the potential to manipulate data structures ten to hundreds of times the size of main memory available for data.

One of the early software design goals was to simplify the process of adding new BRL operators. Each opertor was to be implemented in FORTRAN as a program overlay. An operator would be required to access as many as three simultaneous data sets (e.g., the arithmetic operators PLUS, SUBTRACT, MULTIPLY AND DIVIDE each required two operands and generate one resultant) within a labeled common buffer of 8192 integer values (or 4096 single precision values). Within the code that implements a new operator, files are accessed with a complete set of virtual file handlers coded in FORTRAN. These routines provide services for opening an existing file, creating a new file, reading a vector into the buffer from disk, writing a vector from the buffer to the disk and closing a file. To minimize disk access time and thereby insure optimum response time to the user, techniques employed in managing other virtual memory systems were adapted to the Bugsystem. All data sets, stored on disk as contiguous files, are accessed directly using multiple block transfers. If, for example, a command is issued from an applications program to read a given vector within a file, then the software first determines if the vector is resident within the buffer. If it is resident, then the routine immediately returns to the calling program providing the length of the vector and a pointer into the buffer to the first element of the vector. If it is not resident, its logical address within the file is computed and it--and the vectors which succeed it--are read into the labeled common area. A similar algorithm is employed when writing a vector into a file. A disk write is not required unless the output buffer area is full.

Funneling all file input/output through a common set of routines has significant advantages. The development of new BRL operators is simplified insofar as the underlying applications programs do not each separately (and redundantly) require the algorithms needed to manipulate large files. Another advantage lies in the increased portability of the software. Versions of FORTRAN supported by different machines and operating systems vary most markedly in their non-standardized methods of accessing files. Machine and operating system dependencies are thus isolated in a manageable number of software modules, yielding more portable and more easily maintainable software.

SYSTEM BLOCK DIAGRAM

The major components needed to collect and analyze behavioral data are shown outlined in Figures 1 and 2. Figure 1 shows the major components of the minicomputer system that were purchased on this research grant. An explanation of the components in Figure 1 follows. The video tape deck plays back the images of moving organisms of a given experiment into the video preprocessor (Bugwatcher), which renders the image into X-Y outline coordinates and transfers these coordinates at direct memory access (DMA) speeds (microseconds) into the ECLIPSE's memory. The TV monitor is used to monitor the quality of the ordinary and quantized images and thus provides operator feedback to ensure that the equipment settings are correct. A TV camera may be fed directly into the Bugwatcher, but it is more convenient to record the data at another site and bring in the video tape for a "computer session". The audio channel of the tape deck can be used for recording and playing back tones used to indicate a change in experimental conditions (e.g. initiation of a light stimulus).

The Bugsystem is the name of the complete and rather large network of software subsystems that runs as an interactive application program under the RDOS operating system. The Bugsystem uses a 32 K Word memory partition and has built into it support for alphanumeric output on the terminal (Tektronix 4010) or the line printer and graphics output to either the terminal or the CALCOMP 1038 plotter. Examples of the plotter output are included in this paper. The Bugsystem run time software executes from one of the fixed disks and the users have their own removable disks (5 M Bytes each) for storing data. The continuing software development is conducted from another removable disk. Every week the disks are dumped to mag tape for backup (9 track 800 BPI, RDOS dump format) onto one of three tapes that are rotated. Every six months an archive tape is made of the development software, and as often as requested by Bugsystem users for archiving their data onto mag tape.

The equipment shown in Figure 2 is located in the biological laboratory. Organisms to be recorded are placed in a dish or deep well slide and images of their movements are captured on video tape. The TV monitor is present to insure that the recorded images are free from spurious noise. Even though most of its circuitry is not required in this context, the Bugwatcher is also present for the following reasons. First, it shows what the quantized images look like as they are being recorded. This insures that no valuable behavioral observations will be lost because of improper sync or video levels when the tape is replayed during input to the computer. Second, the Bugwatcher has the tone oscillator and detector circuits for recording tones on the audio channel of the tape deck to denote a change in some experimental condition.

SECTION II

COUPLING TO RESEARCH ENVIRONMENTS

IMAGES

The primary input to the Bugsystem consists of digitized video data. If the video images are sufficiently "clean" (i.e., possess high contrast and lack structural complexity), input of data to the computer is accomplished automatically in real time. Video tapes are replayed into the Bugwatcher which compares the incoming video signal to a "video threshold" set by the user. Those points within the image where the video signal crosses the video threshold are displayed on a video monitor. The user adjusts the video threshold to make these points coincide with the outlines of moving organisms and selects a frame rate (ranging from sixty frames/sec to one frame/min) appropriate for the relative speed of the organisms. When the BUGWATCHER INPUT operator is executed, the computer accepts digitized image information from the Bugwatcher at the selected frame rate. Each threshold point is represented in Cartesian coordinates with 8-bit resolution for each of two orthogonal components. A video frame is represented within the resultant data structure as a data vector with a variable number of such points as its data elements. An entire record (or "video file") consists of a temporally ordered sequence of such vectors.

The Bugsystem was originally developed for the investigation of the behavior of motile microorganisms (3). In this application, the organisms are viewed under dark-field illumination swimming within a well slide upon the stage of a compound microscope. However, automatic digitization of data is possible for any study of moving objects for which "clean" video records are available. One of us (Wilson) is currently using the second generation system to study the effects of plane polarized light upon the behavior of aquatic arthropods (*Daphnia*). The animals move freely within a cylindrical aquarium (diameter ≈20 cm) under bright-field illumination and are viewed using a macro lens attached to the television camera. Use of video tape as a storage medium allows experiments to be conducted in a laboratory remote from the site at which the data are analyzed.

Occasionally video recordings are not "clean" enough to allow fully automated digitization (e.g., data collected in the field) or the digitized images are too crude to provide information about details of an organism's anatomical structure (e.g, the orientation of its eyes). A technique has been developed to expedite manual analysis of such data. Using the PICK operator, the video tape is examined frame-by-frame. The user selects

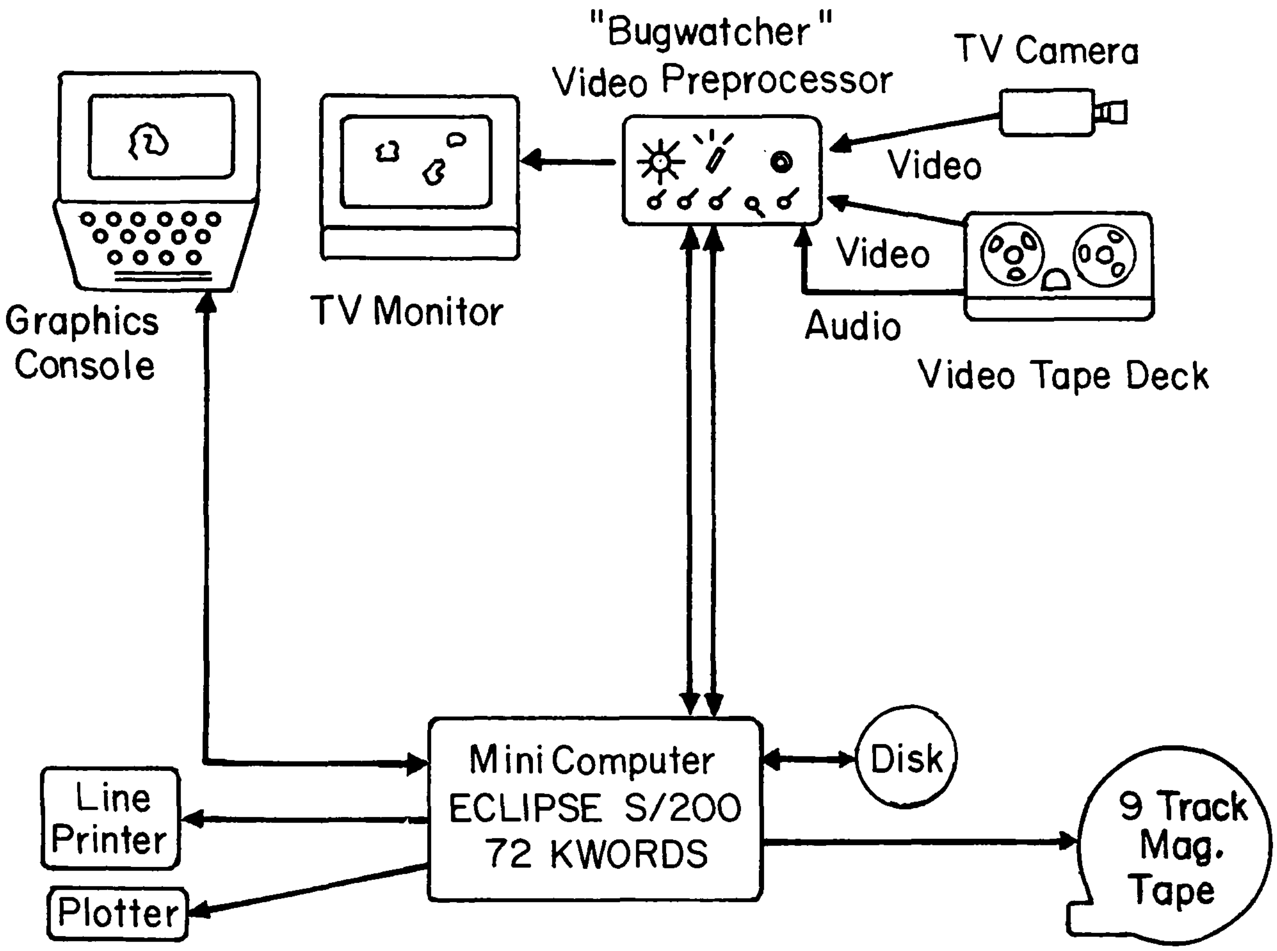


Figure 1. Block diagram of major components of the "Bugsystem".

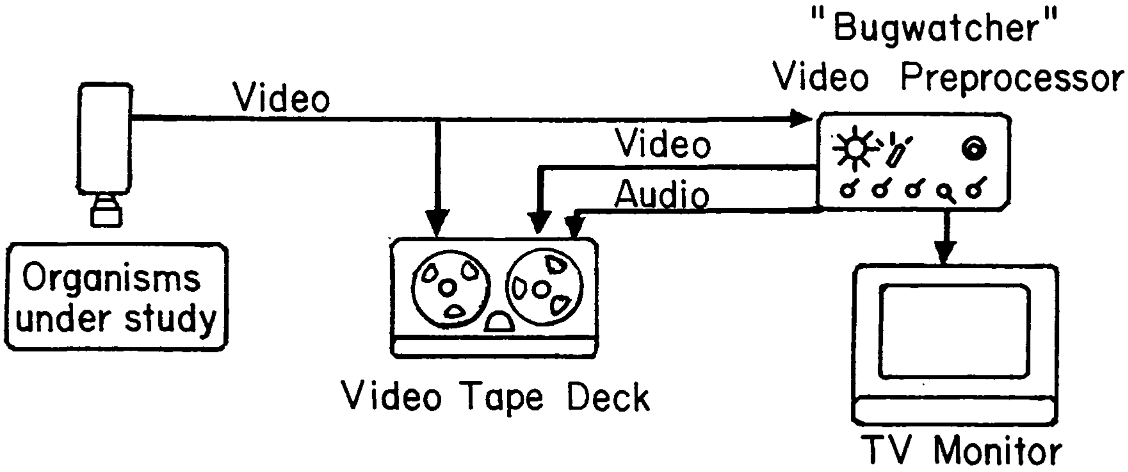


Figure 2. Apparatus used in collecting data in the biological laboratory.

points upon the screen of a video monitor using a video cursor controlled by a joystick. A synthetic video signal representing a tiny bright dot is sent to the Bugwatcher which, in turn, sends digitized video information to the computer. Because averaging algorithms are employed, this method affords higher spatial resolution than does fully automated input: each coordinate in the resultant data structure may have 8-9 significant bits in comparison to the 7-8 bits of the normally digitized input data. This is, of course, orders of magnitude slower than automatic collection of clean images; consequently, manual specification of points through this process is avoided whenever possible.

Input of video data to the computer effects a mapping of the image as seen upon a television screen into a two dimensional representational space. Distances within this space ("Bugspace") are measured in arbitrary internal units (or "Bugwatcher units") and, therefore, must be scaled. Using the LIVE INPUT operator, the Bugwatcher processed image of a centimeter ruler (or any object of known length)--recorded under the same conditions used in gathering behavioral data--is displayed on the screen of a CRT terminal and two points on the rule are selected using the terminal cursors. Thus, the ratio between the distances separating the points in Bugspace and in physical space yields a factor of proportionality between Bugwatcher units and conventional units of spatial measurement (e.g., m, mm, cm, etc.).

EXPERIMENTAL PARAMETERS AND EXPERIMENTAL VARIABLES

Biological interpretation of behavioral data requires that the behavior of organisms be related to experimental conditions prevailing at the time of observation. Consequently, we have developed methods to associate experimental parameters and experimental variables with sets of data. An experimental parameter is a quantity characterizing a condition which is constant throughout any given record but may vary from record to record (e.g., temperature, concentration of a pollutant, etc.). Using the PARAMETER operator, such numerical constants may be inserted into the set of attributes belonging to each data vector. Parameters may be deleted, modified or listed. They may be used to organize graphical displays (e.g., average linear velocity as a function of copper concentration as illustrated in Figure 7) or to modify the execution of certain operations.

An experimental variable is a quantity whose value changes during a single record. Times at which simple step changes in stimulus conditions (e.g., switching a light on and off) occur may be indicated by the presence or absence of tones stored on the audio track of a video tape. The Bugwatcher possesses four tone generators to produce such temporal markers during the course of an experiment; it also possesses external connections which allow the simultaneous actuation of laboratory apparatus. When the video tape is replayed into the Bugwatcher, these tones are detected. As noted above, the second attribute in each data vector sent to the computer contains four bits dedicated to representing the presence or absence of the four tones.

We are presently developing a technique which provides for the representation of stimuli which vary continuously over time. The stimulus level will be encoded by means of frequency modulation (FM) on the audio track of the video tape. When the tape is replayed into the Bugwatcher, the FM signal will be digitized and represented, with 10-bit precision, by the fourth attribute of each data vector sent to the computer. The APPEND operator will be employed to extract this information from each data vector, scale it and store as an equispaced time series. Again these data may be used to organize graphical displays or they may enter into computations involving time series of behavioral variables. One of us (Wilson) is preparing to employ this method to investigate behavior evoked by rotation of the plane of polarized light. The organisms swim beneath a Polaroid filter whose angle is controlled by an analog servomechanism. The filter may be rotated so as to describe quick jumps, ramps, harmonics, etc. A signal directly proportional to the angle of the filter will be encoded and analyzed.

THREE DIMENSIONAL IMAGES

At the time of this writing (summer of 1978), there have been no three dimensional images successfully processed on this system. There is progress in this direction and this section will describe the approach being taken to solve the general problem of tracking fish in X-Y-Z space in a four-foot diameter tank.

The experimental system employs two stationary television cameras mounted approximately 120 cm above a fishtank 120 cm in diameter and 100 cm deep. The cameras are positioned to point to the center of the water surface in the tank and are thus fitted about 25° away from the normal. It was necessary to reverse the direction of the Y-scan on one of the cameras so that the image viewed on two monitors would both have the "top" of the picture be the same general location in the fish tank. Without this reversal, which was accomplished electronically, one picture appeared "upside down" with respect to the other.

The goal of this 3-D processing system is to use the same general Bugwatcher hardware and software system to analyze motion in three dimensions. Of course, we foresaw the development of special 3-D operators to generate and interpret triads (rather than pairs) of data values as well as some special hardware to enable analog recording and digital input of such data. For example, data are recorded using a specially built video multiplexer called the J-Box (after its creator, Jay Sinnett) which multiplexes--or alternately switches between--the two cameras to produce a single video output that can be recorded on a conventional tape deck. Thus, the maximum effective frame rate is reduced from 60 to 30 f/sec for stereo images. Nevertheless, the video data produced in this manner can be entered into the computer automatically using the existing BUGWATCHER INPUT operator. This presupposes that the illumination of the fish in the tank is designed and adjusted to yield high contrast (i.e., "clean") images as required by the Bugwatcher for automatic input. Consequently, stereo images are input to the Bugwatcher as is done with a normal single image video recording. Likewise centroids can be computed using existing routines and the software is oblivious to the fact that alternate frames were fish generated from two different views of the same scene. These stereo mode centroids are input to the *MAP3* operator which converts all of the X-Y points into corrected water surface coordinates, and, using alternate frame and camera position information, generates three dimensional positions of the fish centroids. The *PAT3* operator connects these centroids into continuous path trajectories in 3-space. With the data organized in this way, it is not difficult to generate time series functions of the path data using operators such as *LIN3* for 3-D linear velocity and *DIS3* to compute the continuously varying distance of a fish to its nearest neighbor.

Improperly functioning hardware and failure to obtain high contrast video images of the fish swimming against a uniformly illuminated background proved to be major stumbling blocks in the analysis of 3-D data. Unfortunately, the grant period ended before any real behavioral data could be analyzed using this system. However, as of this writing, the prospects look reasonably good that the system will be utilized to track organisms swimming in three dimensions. But due to a lack of clean stereo images, the automated input of video data--a technique which has been very successful in investigations of two dimensional patterns of movement exhibited by organisms swimming within smaller containers--has been abandoned. In a recent attempt to image fish swimming within the tank as distinct dark objects moving against a uniformly bright field, the fish appeared as dark gray entities moving across a light gray to dark gray background. Therefore, the analysis of such stereo images will entail using the manual (or semi-manual) data input procedure implemented via the *PICK* operator: the video tape must be stepped through manually using the still frame control on the video tape playback unit until the desired data records have been collected. Thus obtained, the analysis of these data will proceed as outlined above, through application of the *MAP3* and *PAT3* operators. It should be noted that the Bugsystem still greatly facilitates the quantification of the behavior of organisms moving freely through a three dimensional space. It provides a fairly accurate means of measuring spatial relationships of motile organisms over time. Although the data collection process is much more time consuming as compared to the automatic collection of two dimensional high contrast data, subsequent investigation can proceed with all of the available analytic tools such as interactive graphics, mathematical manipulations, and statistical analyses developed for such data.

SECTION III

PROCESSING AND DISPLAY

IMAGE PROCESSING

As discussed above, input of video 2-D data to the computer entails substantial preprocessing of pictorial information. A data vector within a resultant video file is a list of contemporaneous points, with an organism's outline represented within this data structure as a localized set of points. The user can display such data graphically (using the PLOT operator) or alpha-numerically (using the LIST or EXAMINE operators). Video files may be edited both in time and in space. The EDIT operator allows one to save or delete temporally contiguous sets of data vectors. Thus, the user could EXAMINE the data to ascertain the frame at which the status of a tone had changed (indicating a change in stimulus conditions, e.g., switching on a blue light) and then EDIT the data to insure that this change occurs on frame number 100. The MASK operator allows one to save or delete points within rectangular or circular regions of the image plane. Thus, the user could MASK out all points within a video file which correspond to a particle of detritus within the experimental preparation. Finally, the user may APPEND additional information to a video file (e.g. text describing the conditions of the experiment, numerical constants, time series of tone states or time series of experimental variables).

Analysis of video data by means of the Bugsystem proceeds by abstracting one (or more) points from each point set which delineats an organism's outline. In an investigation of translational movement this task is easily defined. Unlike either rotational or flectional movement, quantitative description of the translational component of an individual organism's behavior does not require detailed knowledge of the organism's external anatomy. The body of the organism is represented by a single point, viz., its "center of mass". Translational movement is defined as displacement of this point from one position in space to another.

Mapping outlines into centrally located points is usually achieved by means of the CENTROID operator whose command syntax is exemplified by the entry

```
*CENT BUGS.VI BUGS.CE n1, n2, n3.
```

Each vector in the resultant file "BUGS.CE" corresponds to a vector in the operand file "BUGS.VI". Each element of a resultant data vector is a "centroid": a point in Bugspace whose X and Y coordinates are, respectively, the average X and Y coordinates of an "acceptable set" of points in the corresponding operand data vector. The numerical parameters "n1", "n2" and "n3" are required to characterize an "acceptable set" of operand data points. An initial member (having non-zero coordinates) of such a set is chosen from the operand data vector, the coordinates of this point are set equal to zero in the operand data and a mask (width = 2·n1, and height = 2·n2) is centered upon this point. The set is augmented if an operand point (having non-zero coordinates) falls within the mask. The mask is then centered upon the new point and the search continues. The search terminates when the set can no longer by augmented; the set is deemed "acceptable" if it possesses at least n3 members. Ideally, this process yields one centroid corresponding to each outline--unless the outlines of two organisms are merged. The user can PLOT the centroids over the original data to confirm this correspondence and recompute the centroids using new parameters if the correspondence is not adequate. The structure of "centroid data" is quite similar to that of video data except that each coordinate of a centroid is represented with 15-bit (rather than 8-bit) precision in attempt to exploit the greater accuracy resulting from the averaging procedure. Centroid data is displayed and edited in the same fashion as video data.

The PICK operator provides an alternative method to abstract points of interest from video data. The use of this operator for direct analysis of a video tape, thereby bypassing the acquisition and processing of video files, is discussed above (see "COUPLING TO RESEARCH ENVIRONMENTS"). The user may also PICK points associated with each outline by means of the terminal cursors. The PICK operator is indispensable for investigations of rotational and flectional movement. For example, the user might elect to study the orientations of the longitudinal body axes of a group of organisms (i.e., simple rotation in a single plane). The longitudinal axis may be defined as a vector extending from the tail to the head of an organism. Thus, the user picks an ordered pair of points corresponding to each outline by entering

```
*PICK/VI BUGS.VI BUGS.TH
```

and using the cursors to specify first the "tail" and then the "head" associated with each outline. The resultant file "BUGS.TH" possesses the structure of centroid data with each "tail point" immediately followed by the correlated "head point". These points are then segregated using the PLUCK operator. For example, the two commands

```
*PLUC BUGS.TH BUGS.T 1,2
*PLUC BUGS.TH BUGS.H 2,2
```

respectively produces the files "BUGS.T"--containing all first elements of the ordered pairs (viz., the "tail points")--and "BUGS.H"--containing all second elements of the ordered pairs (viz., the "head points"). This type of analysis may be extended to encompass larger collections of points (or n-tuples), thereby providing for the study of flectional movement (e.g., the angles of propulsive appendages with respect to the longitudinal axis).

Regardless of the type of movement under investigation, the next stage of image processing entails the compuation of paths or trajectories through Bugspace. A path or trajectory is a time-ordered set of points represented as a single data vector and characterized as follows: all points in a path are selected from an operand file consisting of centroids or having the structure of centroid data. The manner in which points are represented in a path is identical to the way they are represented in a frame of centroid data. Each path starts within a specific frame, with its starting frame number represented as the fourth attribute of the vector. Over the temporal interval during which a path is defined, one (and only one) point is selected from each corresponding vector (or frame) of the operand file. Adjacent pairs of points within a path are selected on the basis of their spatial contiguity within adjacent frames.

The search for paths is performed by the PATH operator whose command syntax is exemplified by the entry

```
PATH BUGS.CE BUGS.PA n1, n2, n3, n4, n5.
```

The numerical parameters "n1" through "n5" control various details of the search. n1 specifies the width (in Bugwatcher units) of a square mask used as a criterion of spatial contiguity. n2 is the maximum number of times to expand the mask if no contiguous point is located within the frame being searched. n3 is the minimum number of points to be accepted as a valid path. n4 is the minimum average displacement in Bugspace between consecutive frames for a set of points to be accepted as a valid path; this parameter may be used to "weed out" stationary artifacts (e.g., a particle of detritus). n5 is the maximum number of frames to "look ahead"; i.e., if no contiguous point is found in the current frame, then the next frame may be searched. If a point within the new frame qualifies, the missing point is computed by linear interpolation.

Ideally, every path would correspond to the movement of a single point (associated with the outline of each individual organism) through Bugspace and each such movement would be respresented by one path. In practice, the PATH operator may commit two types of errors. The operator may overlook certain segments of continuous movement. Such omissions may yield an abnormally short path, or they may result in a one-to-many (even a one-to-none) correspondence between real word trajectories and paths. Alternatively, the operator may confound certain segments of the continuous movements of two (or more) organisms. If there were but one organism in the field of view at any time, then errors of omission could be abolished by using a large mask (why not let it include all of Bugspace?) and allowing the program to look ahead several frames. But with several organisms represented within each frame there is clearly a tradeoff in choosing parameters so as to reduce the incidence of the two types of errors. The user may PLOT the paths over the centroids (or over the original video data) to check their fit and then recompute the paths using new parameters. Often several iterations of this process are required to obtain an optimum combination of parameters. Since the parameters depend largely upon the magnification, the density of organisms and the way in which they move, the same "optimum set" of parameters (retained in memory for the user's convenience) are generally used to compute paths for all replicates of an experiment.

Proper selection of path finding parameters can significantly reduce the number of erroneous paths, but cannot be relied upon to eliminate all errors. Consequently, we have developed programs which allow the user to interactively detect and correct mistakes within path files. Path editing programs (inclucing CHOZ, EDIT, PEDI and JOIN) allow the user to perform the following basic operations: (1) delete a path, (2) truncate a path, (3) cut a path into two smaller paths and (4) join two paths (assuming they do not overlap in time).

Once a valid collection of paths has been obtained, the user may proceed directly to the extraction of time series of behavioral variables (as discussed below) from the path files. Before doing so, however, there are several additional procedures which the user may choose to apply to the path data. Since the use of these procedures (and the order in which they are applied) is dependent upon the overall design of the experiment, we will first illustrate them by means of specific examples.

Wilson video taped the behavior of *Daphnia pulex* (a small freshwater crustacean, commonly known as a "water flea") in Talbot Waterman's laboratory at Yale University. Twelve animals were observed from below swimming against a brightly and uniformly illuminated background. A variable polarizer/depolarizer was interposed between the chamber containing the animals and the light source. The tape consisted of 23 separate experimental records. Before each recording the polarizing filter was rotated to a randomly chosen angle and, also in random sequence, the device was adjusted so as to polarize or depolarize the illumination. Each record began with the image of a strip of plastic attached to the filter to indicate the angle of the filter and the magnification of the image. The plastic strip was then removed and the animals were observed swimming under constant conditions. In all, 13 records were obtained under polarized light and ten records were obtained under unpolarized light. Each recording lasted two minutes.

The video tape was analyzed with the aid of the Bugsystem at Narragansett, R.I. For each record the LIVE INPUT operator was used to determine the angle of the filter with respect to the Bugsystem reference frame (which is fixed with respect to the raster scan of the video signal). As discussed above, the appropriate spatial scale factor (approximately 0.40 mm/Bugwatcher unit) was also computed. The BUGWATCHER INPUT operator was then used to generate one video file (480 frames of data at five frames/sec or 96 sec of data) for each record on the tape. Wilson then MASKed the data, saving only those points within a centered circular region of Bugspace. The centroids and then the paths were separately computed for each masked video file using the same set of centroid and pathfinding parameters. Very few pathfinding errors resulted from these computations, but these were corrected by editing the separate path files. Wilson then employed the PARAMETER operator to associate the angle of the filter with respect to the Bugsystem reference frame (an experimental parameter) with each path. This angle was the same for every path within a given path file. By repeated use of the MERGE operator, Wilson condensed the data to produce two exceedingly large files: one containing all paths observed under polarized light (154 paths, 13210 data points) and another containing all paths observed under unpolarized light (166 paths, 13691 data points). These paths were still represented in Cartesian coordinates relative to the Bugsystem reference frame. However, orientation with respect to the plane of polarization should only be manifest with respect to the reference frame of the filter. Using the ROTATE operator, the two frames of reference were made to coincide: every path within each merged file was rotated in Bug-space (about an axis passing through the center of the space)through an angle obtained by negating the appropriate experimental parameter for each path. All resultant paths for *Daphnia* swimming under both polarized and unpolarized light are PLOTed in Figure 3. As shown in Figure 4, many of these paths are aligned approximately orthogonal to the axis of the filter (i.e., perpendicular to the *e*-vector) in the presence of polarized light.

Drs. William Lang and Don Miller, Sally Lawrence, Martha Marcy and Anne Ackenhausen have employed the Bugsystem to conduct preliminary studies on the effects of copper exposure upon swimming behavior of barnacle larvae (viz., second stage nauplii of *Balanus improvisus*). The nauplii were exposed to six different levels of copper for 24 hours prior to observation: 3.2, 25, 60, 115, 147 and 205 ppb. No significant mortality resulted from such exposure. They viewed the larvae with the aid of a disecting microscope using 830 nm substage illumination and obtained video tapes of larval behavior in "darkness" (i.e., stage illumination only) and when stimulated with 488 nm light (0.06 W/M^2). Three replicates were performed for each experimental condition (36 observations in all). These data were digitized at 10 frames per second. Photic stimulation was indicated as the Bugwatcher detected a tone denoting the onset of 480 nm illumination. Video files for stimulated nauplii were EDITed to start 0.50 seconds after the light was turned on and to last 2.00 seconds. For all files centroids and paths were computed, then three path files corresponding to each experimental condition were MERGEd to form a single file. Unfortunately, the computed trajectories of larvae exposed to high levels of copper (150 ppb) were quite ragged and irregular due to the low linear velocities of the larvae. Consequently, all path data were SMOOTHed using a linear low-pass filter: the SMOOTH operator computes moving average of X and Y components with weights as specified by the Tukey window (2). Smoothed paths of barnacle nauplii are shown in Figure 5.

The preceeding analyses illustrate the use of operators to organize path data *prior to* the computation of behavioral variables. The ROTATE operator was implemented to expedite studies of animal orientation. Using this operator, all spatial data in a file may be rotated through a constant angle, each path may be rotated through a constant angle associated with

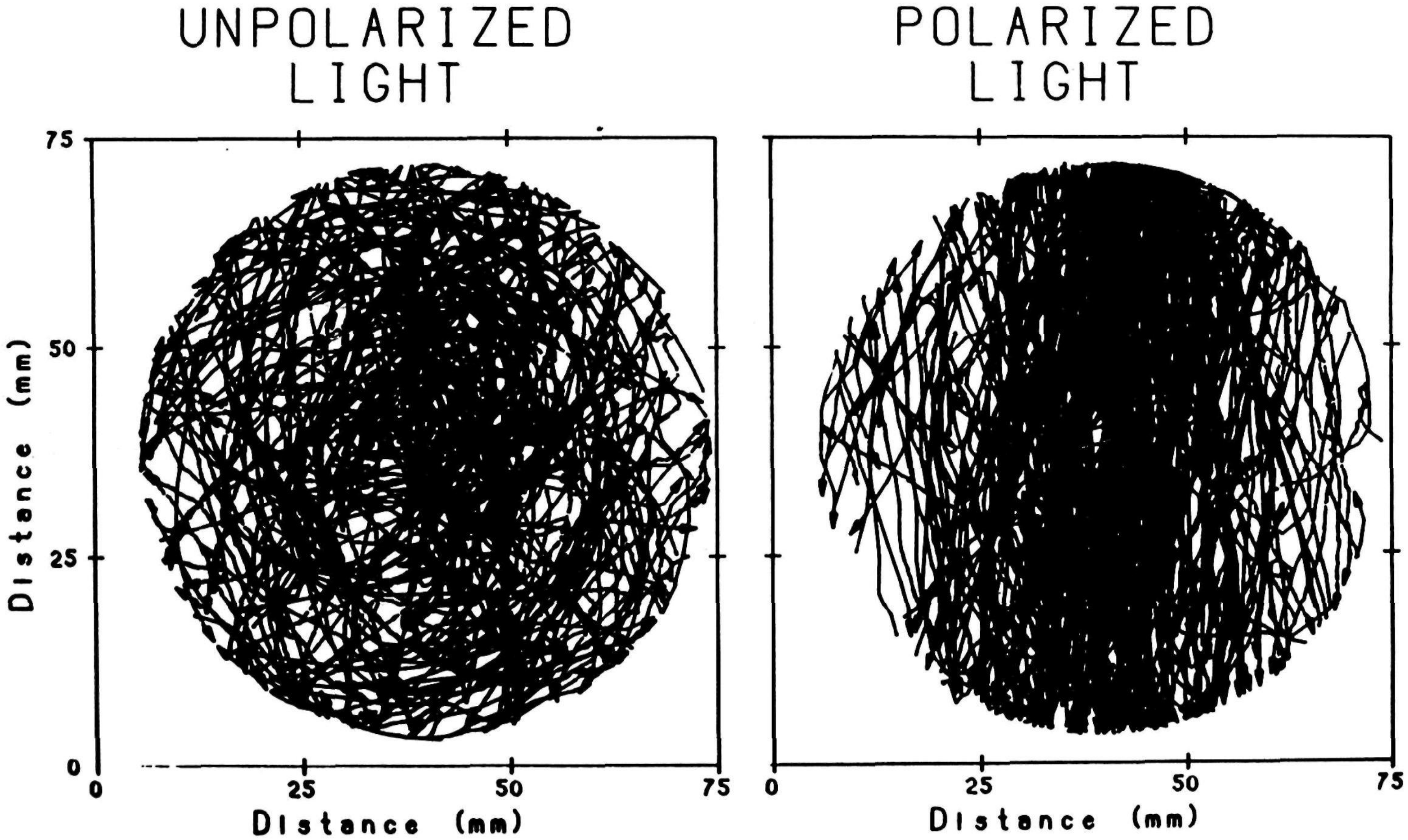


Figure 3. Polarotactic orientation of Daphnia pulex as illustrated by merged and rotated paths in unpolarized (166 paths) and polarized (154 paths) light. The sequence of operations required to generate these data are discussed in the text.

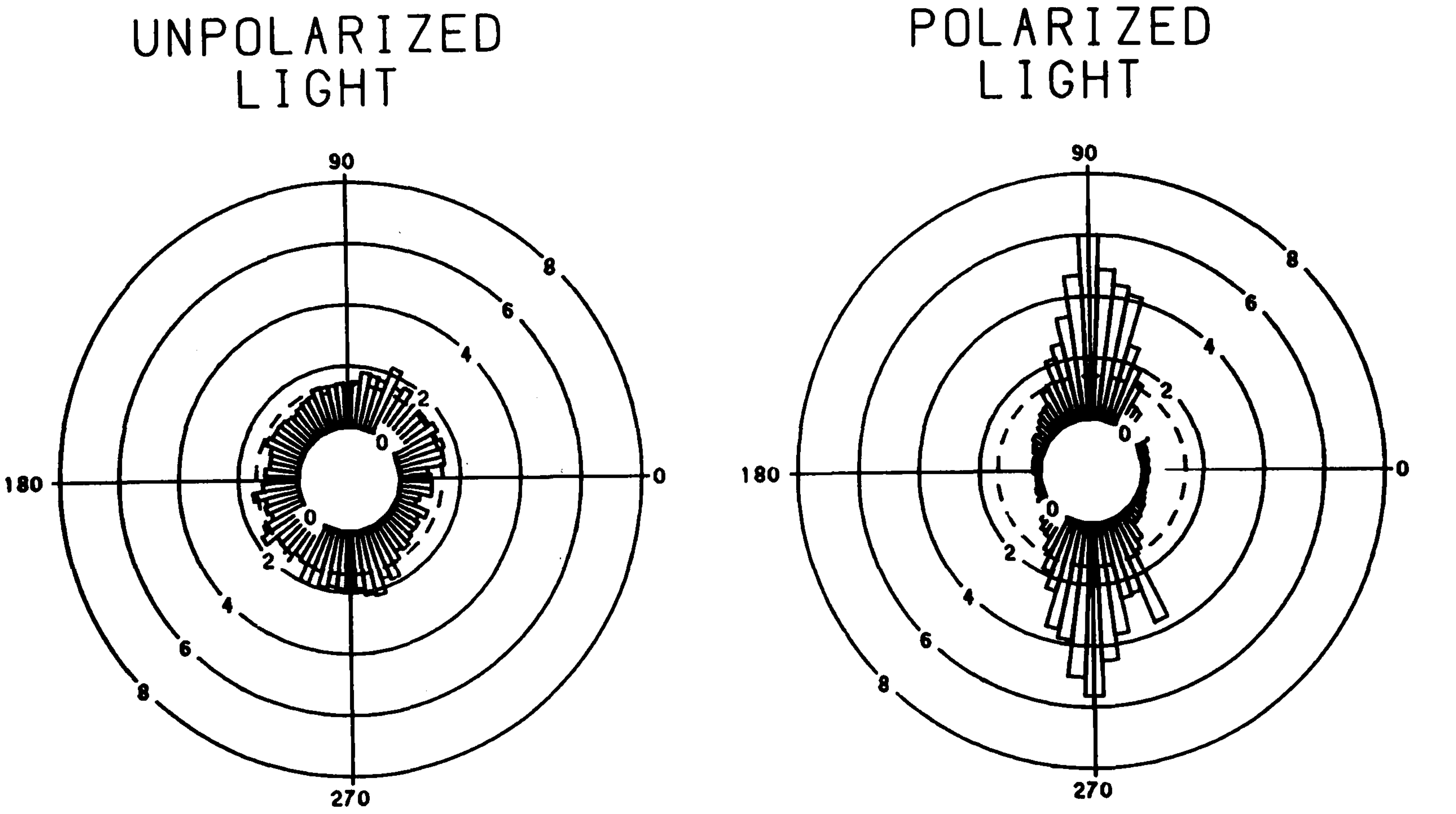


Figure 4. Polarotactic orientation of *Daphnia pulex* as illustrated by polar wedge histograms of instantaneous direction of travel in unpolarized (13691 data points) and polarized (13210 data points) light. Percent of sample is plotted on the radius and the dashed circle denotes the expectation for a circular uniform distribution. The e-vector of the plane polarized light is horizontal (viz., 0 and 180 degrees) in the graph on the right.

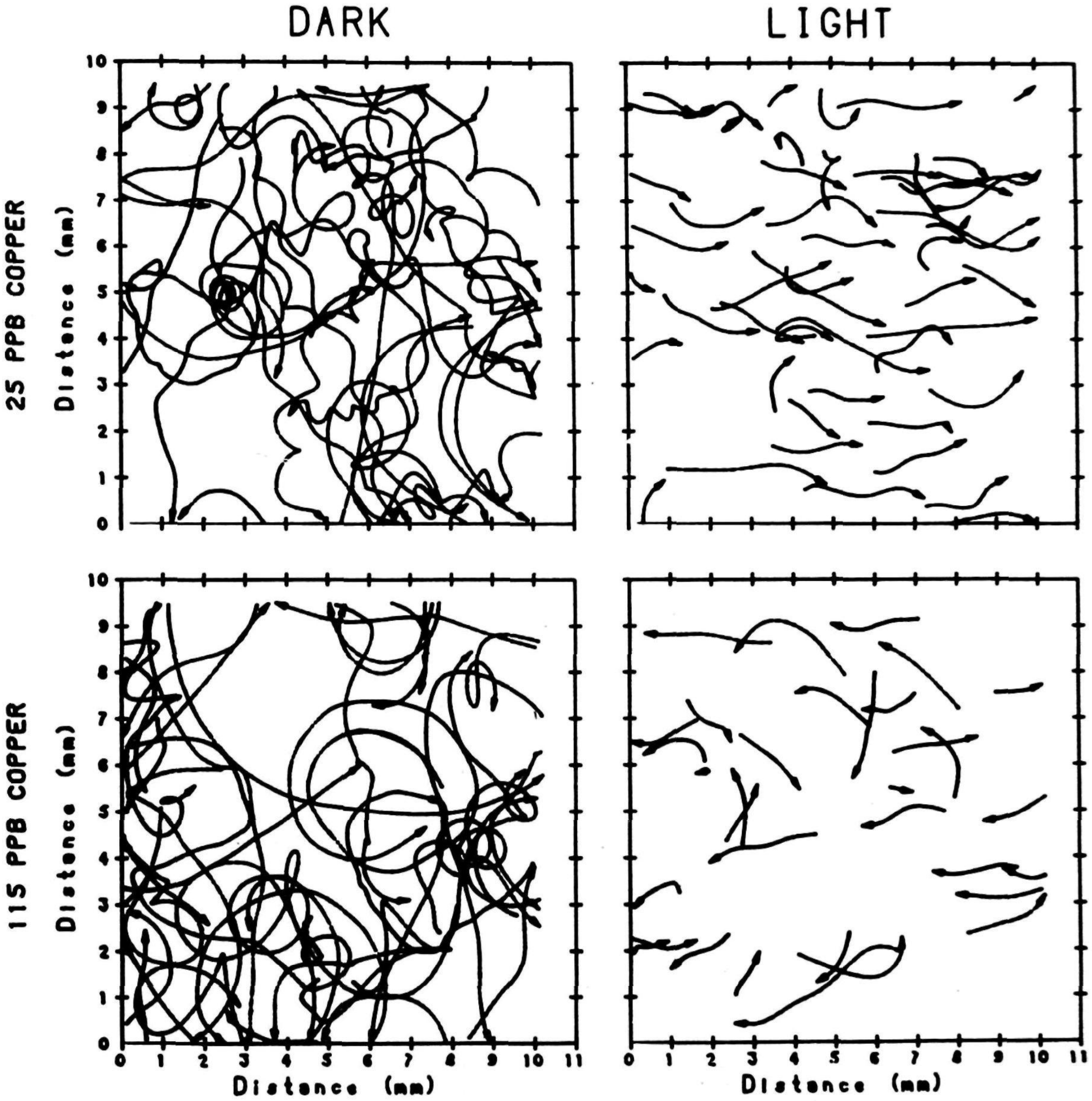


Figure 5. Merged path files of Balanus improvisus nauplii (barnacle larvae) in darkness (column on left) and stimulated by 480 nm illumination (column on right) at two different levels of copper exposure (top row, 25 ppb Cu; bottom row, 115 ppb Cu). Control records are approximately 35 sec in duration. Experimental records begin 0.5 sec after stimulus onset and last 2.0 sec. Photic stimulation was aligned horizontally, originating from the right side of the above plot.

that path or each point may be rotated through an angle associated with a corresponding moment in time. The last option enables the investigator to study orientation with respect to a moving stimulus. Files may be MERGED whenever they are taken to be replicates of the same experiment. Not only does this simplify the bookkeeping tasks associated with subseqent analysis, but, in addition, allows a set of similar data to be treated as a single sample by statistical operators. The fundamental advantage of an operator-based interactive system for the analysis of behavioral data is its flexibility: the operators which the user chooses to apply to the data--and--the order in which they are applied can be selected to correspond to the design of the original experiment.

GENERATING AND TRANSFORMING TIME SERIES DATA

Time series of X and Y coordinate values are generated from path data by means of the SPLIT operator, whose command syntax is illustrated by the entry

```
*SPLI BUGS.PA BUGS.X BUGS.Y.
```

Each data vector in the resultant files "BUGS.X" and "BUGS.Y" is, respectively, a time series of X and Y coordinate values. The resultant data sets are in one-to-one correspondence and each element is represented as a single precision floating point number. Like the paths from which they are derived, resultant time series may start and end at arbitrary points in time. Therefore, the number of series defined at any given moment is also arbitrary. Other computer systems have been developed for the anlysis of equispaced time series (8). After image processing has been completed, the operators available on the Bugsystem are unique only insofar they possess the sophistication required to manage large collections of arbitrarily derived series (e.g. Figure 6).

In principle, all behavioral variables which may be investigated using the Bugsystem can be generated by simple arithmetic transformation of series of X and Y coordinates. Greaves (5,6) has discussed the computation of certain behavioral variables (viz., linear velocity, net to gross displacement ratio, direction of travel and angular velocity) using arithematic operators implemented within the prototype system. Such operators transform every element of every data vector within a file, treating each vector as a separate unit of data. The present Bugsystem is also provided with a wide selection of simple arithmetical operators. However, we have condensed the computation of certain frequently calculated behavioral variables into single operators which require path data as input. As an example, let us consider the RATE OF CHANGE OF DIRECTION operator whose command syntax is illustrated by the entry

```
*RCDI BUGS.PA BUGS.RD n1.
```

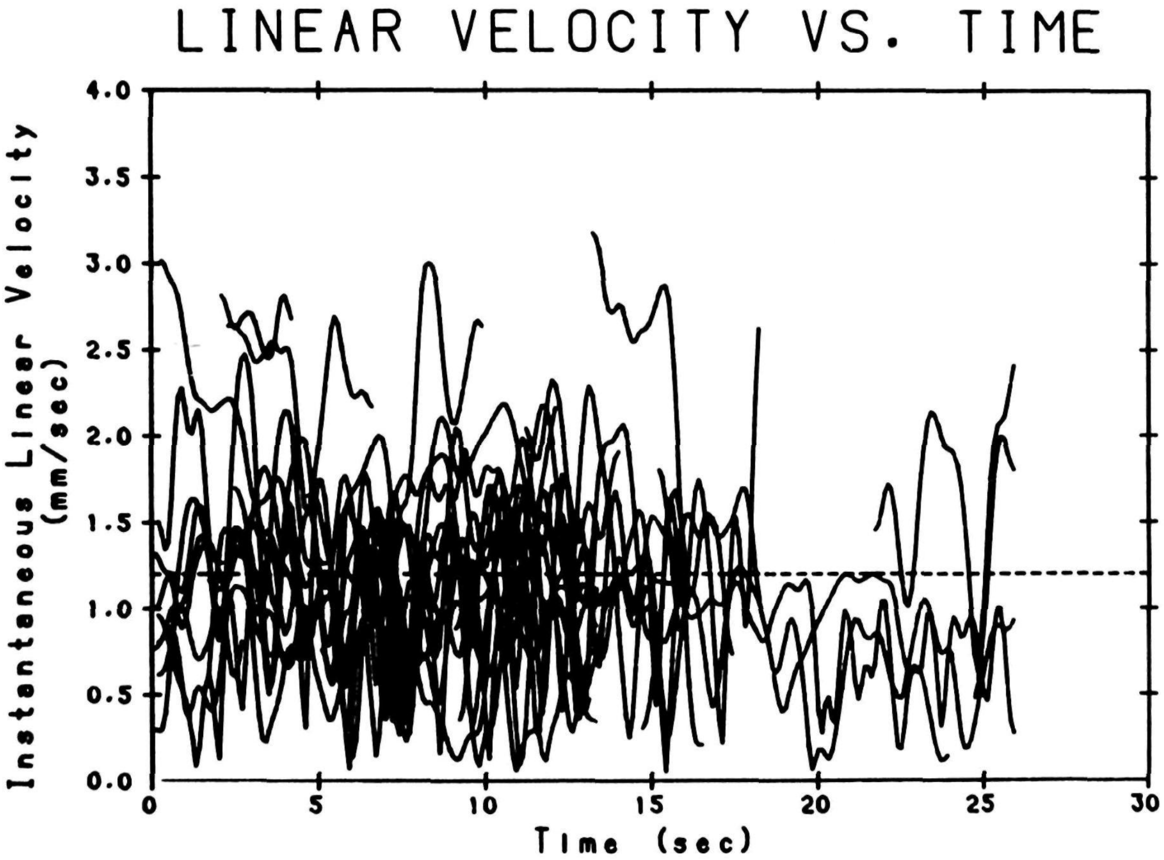


Figure 6. Linear velocity as a function of time for B. improvisus nauplii exposed to 25 ppb copper in the absence of photic stimulation. Sample consists of 37 paths (2369 data points); the dashed line denotes the estimate of the mean (1.20 ± 0.01 mm/sec).

Every element in the resultant file "BUGS.RD" is the (unsigned) rate at which the corresponding path changed its direction of travel at a given moment. RCDI is the absolute value of angular velocity expressed in degrees per second (The parameter "nl" is the frame rate, requried to convert from degrees per frame.) The importance of this variable in ascertaining mechanisms responsible for certain animal aggregations has been widely discussed (4). RCDI may also be computed by a sequence of simple arithmetical operations as follows:

```
(a)  *SPLIT BUGS.PA BUGS.X BUGS.Y
(b)  *CDIF BUGS.X BUGS.DX
(c)  *CDIF BUGS.Y. BUGS.DY
(d)  *CTAN BUGS.DX BUGS.DY BUGS.DI
(e)  *CDIF BUGS.DI BUGS.AC
(f)  *MULT/CO BUGS.AC BUGS.AV nl
(g)  *ABSV BUGS.AV BUGS.RD
```

Where "CDIF" if the CENTRAL DIFFERENCE operator (a discrete approximation to the differential operator), "CTAN" is the CONTINUOUS ARCTANGENT operator, "MULT/CO"denotes multiplication by a constant and "ABSV" is the ABSOLUTE VALUE operator. The resultant "BUGS.DI" of step (d) is the direction of travel measured in degrees with respect to the Bugsystem reference frame; it could have been generated from the original path data using the DIRECTION OF TRAVEL operator. The resultant "BUGS.AV" of step (f) contains angular velocities (measured in degrees per second); this file could have been produced using the ANGULAR VELOCITY operator. Other operators have been developed to evaluate LINEAR VELOCITY and NET TO GROSS DISPLACEMENT RATIO functions defined upon path data.

SIMPLE STATISTICAL PROCESSING

For the purpose of statistical analysis two different types of data structure--representing two levels in a structural hierarchy--may be distinguished as "samples": vectors and files. Many statistical operators recognize this distinction. For example the STATISTICAL PARAMETER operator estimates parameters such as the mean, variance, standard deviation, skewness, kurtosis, etc.

The command

```
STAT/VE BUGS.LV BUGS.ST
```

produces the resultant file "BUGS.ST," containing one data vector (i.e., a list of statistical parameters) for each data vector in the operand, whereas the command

```
*STAT/FI BUGS.LV BUGS.ST
```

produces only one resultant data vector characterizing the entire file. In either case, the user can LIST the resultant parameters. Similary, the SLOT operator provides for estimation of density and distribution functions via histograms both for individual vectors and entire files. These data may be displayed graphically (Figures 4, 8 and 9) or LISTed on the terminal or the line printer.

We chose to store statistical parameters within data vectors, rather than merely computing and displaying them, in order to allow them to enter into subsequent calculations. For example, assume the user had computed statistical parameters for each data vector in a file of instantaneous linear velocities (e.g., the "STAT/VE" example given above); each estimate of the mean is thereby an average for each path. The user may then investigate the distribution of these path averages. The estimated means are first isoloated using the STRIP operator:

```
*STRI/MN BUGS.ST BUGS.MN.
```

The resultant file "BUGS.MN" contains a single data vector; each element of this vector is a mean ("/MN") stripped from one operand data vector. The user may wish to compute statistical parameters for the new set of data, or explore its frequency distribution via histograms. The user may also MERGE such files so that each data vector corresponds to a single experimental condition. The vectors may then be compared with one another (e.g., one-way analysis of variance with the ANOV operator, chi-square tests with the CHI2 operator, etc) or graphically displayed as shown in Figure 7.

We have also made special provision within the Bugsystem for the manipulation and statistical analysis of angular data. For example, the command

```
*STAT/AN/FI BUGS.DI BUGS.ST
```

produces a single data vector whose elements are statistical parameters appropriate for circular distributions. These include the length and direction of the mean vector (and related measures) as discussed by Batschelet (1). Moreover, because the Bugsystem is well suited for investigations of animal orientation we have implemented an extensive polar graphics package within the Bugsystem. Figures 4 and 9 illustrate the use of polar wedge histograms to represent angular density functions.

MORE ADVANCED OPERATIONS

As is illustrated in Figure 6, a typical behavioral record consists of multiple time series which begin and end at arbitrary points in time. We have implemented operations to discern time-varying trends within such collections of data. One approach we have taken is to develop so-called "running" statistical operators: operators which provide estimates of statistical quantities for all points within vectors defined at each frame or instant of time. For example, the RUNNING AVERAGE computes an ensemble

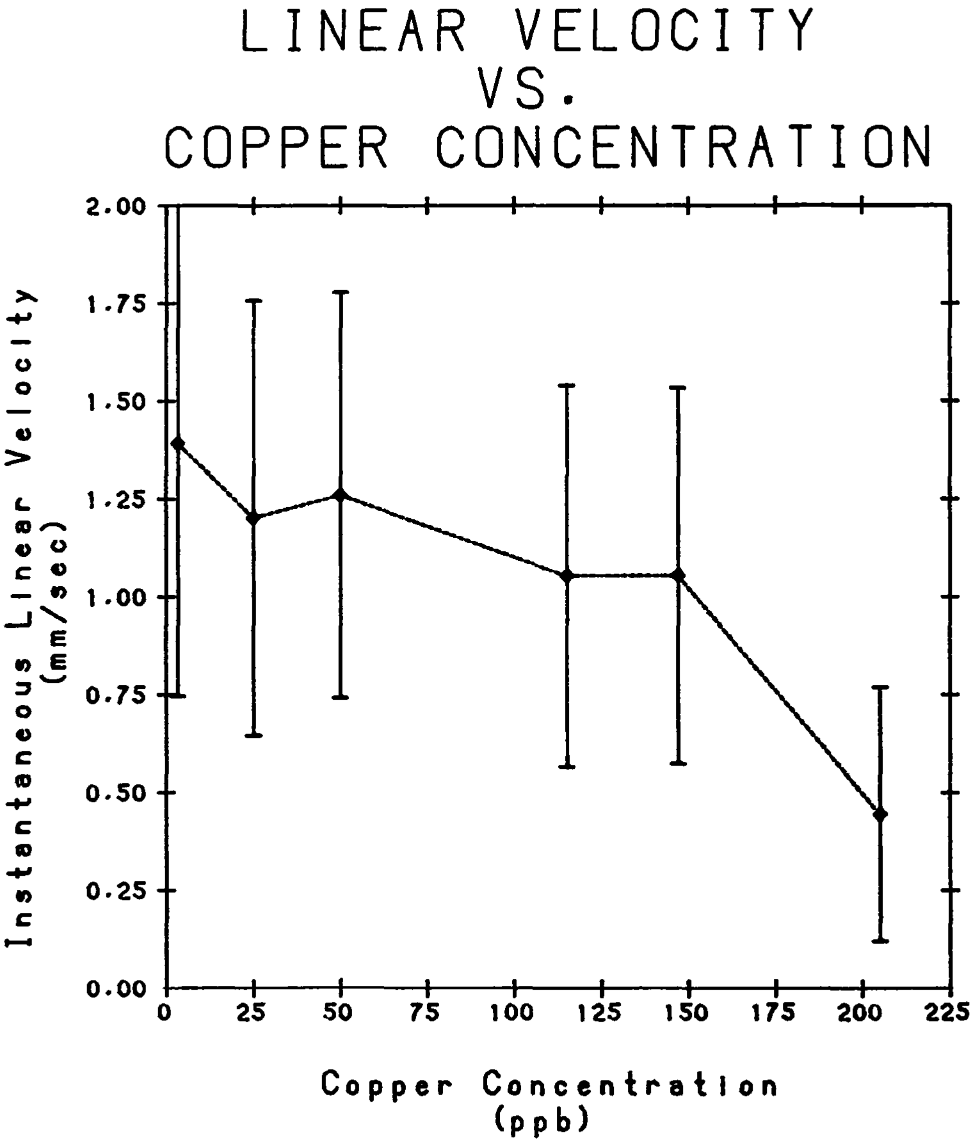


Figure 7. Average linear velocity of B. improvisus nauplii for each sample (± standard deviation) plotted as a function of level of copper exposure in the absence of photic stimulation. The number of instantaneous linear velocity values for each copper concentration ranged between 2055 and 3560. Copper exposure time = 24 hr.

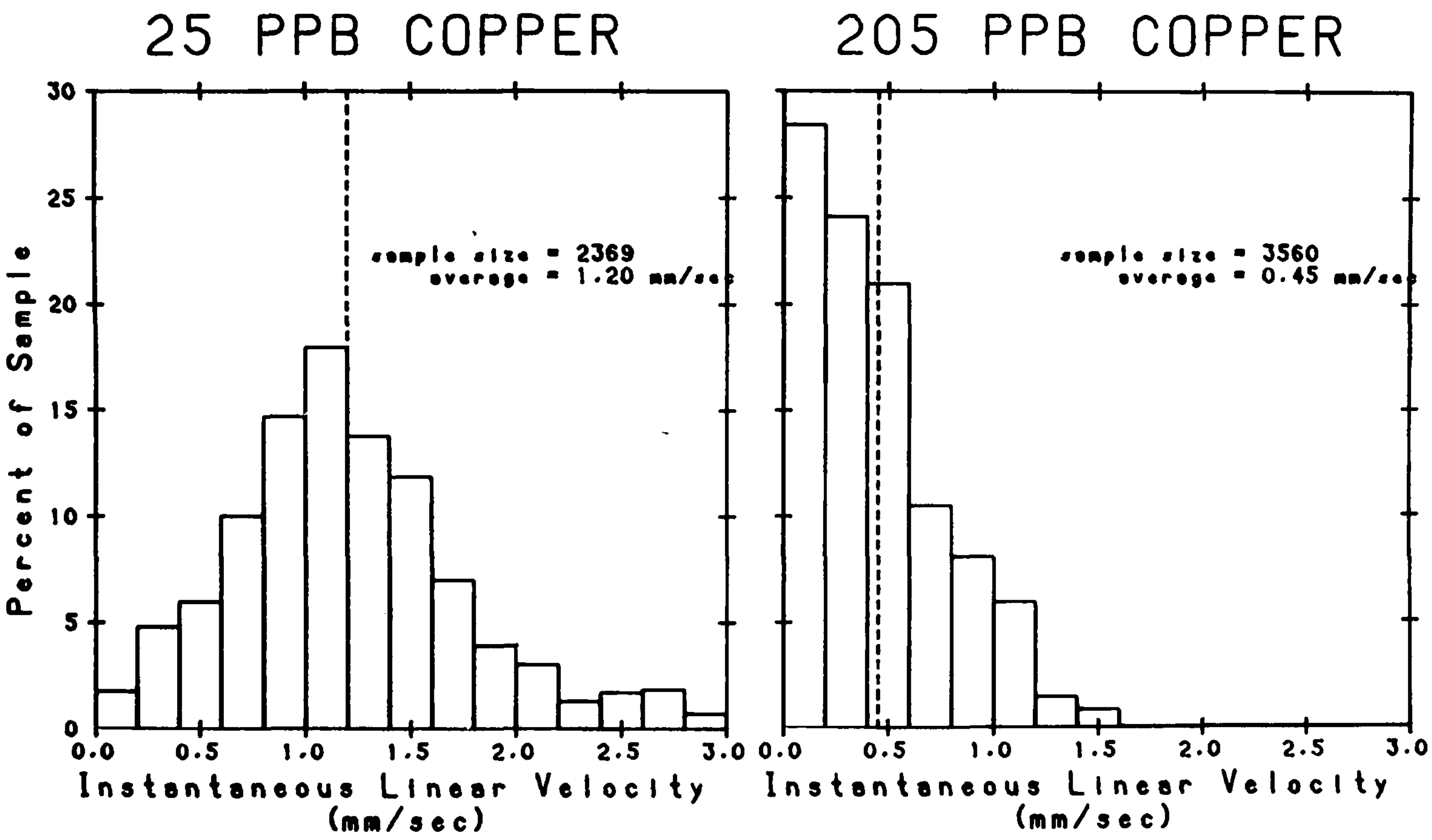
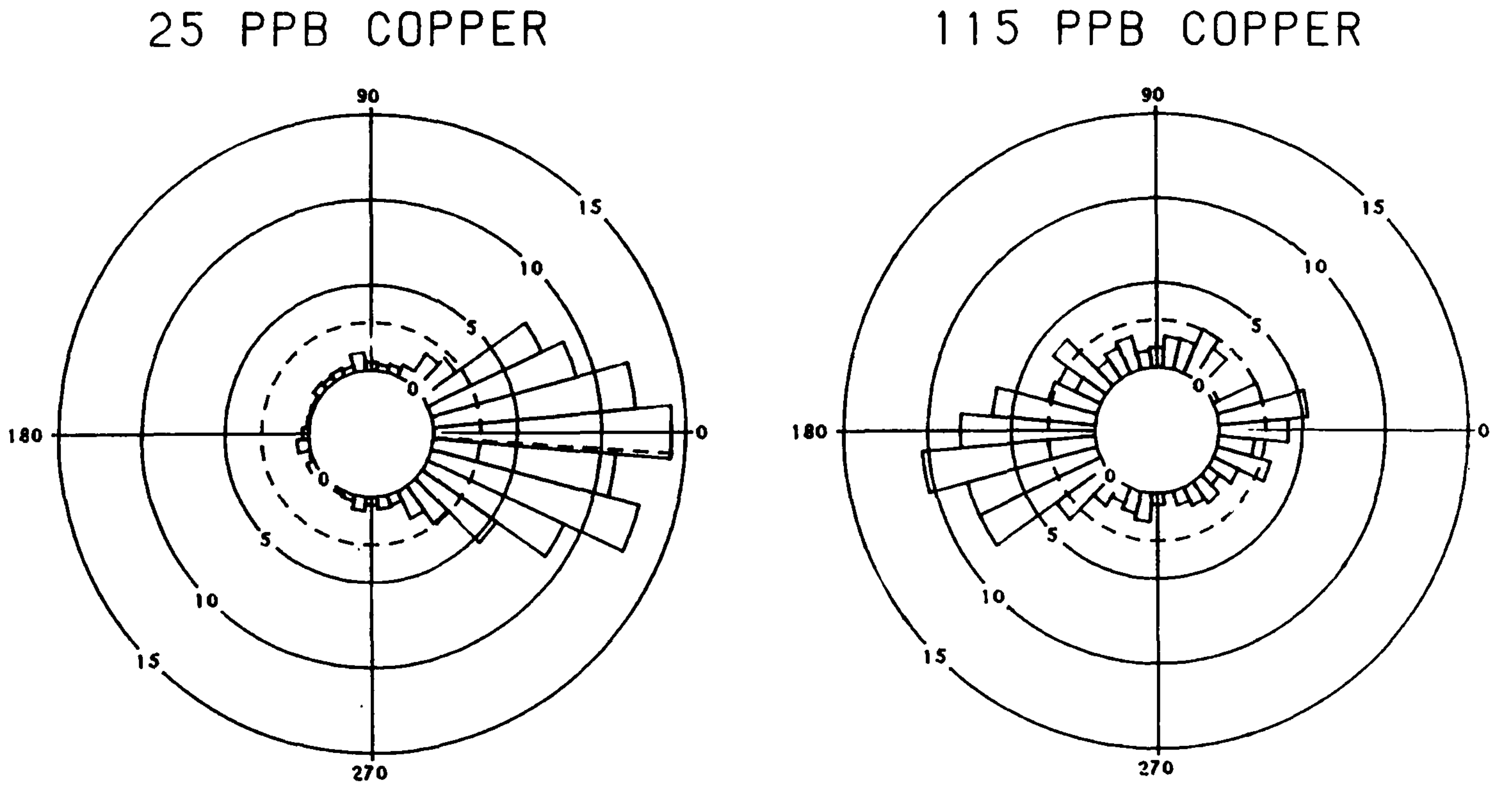


Figure 8. Distributions of instantaneous linear velocity of B. improvisus nauplii at two different levels of 24 hr. copper exposure: 25 ppb (left) and 205 ppb (right); no photic stimulus present. Vertical dashed lines denote estimates of the mean of each sample.


25 PPB COPPER

115 PPB COPPER

Figure 9. Photactic orientation of B. improvisus nauplii elicited by 480 nm illumination at two different levels of 24 hr. copper exposure: 25 ppb (sample size = 854) and 115 ppb (sample size = 598). The angle denotes instantaneous direction of travel with respect to the light source; percent of sample is represented on the radius. The stimulation is directed from the right of each polar graph (viz., 0.0 degrees). The dashed circles denote the expectation for circular uniform distributions.

average for all simultaneously defined operand values, storing the resultant values within a single data vector. Similar routines provide for estimates of ensemble standard error of the mean and direction of ensemble mean vector. We have also developed a RUNNING SLOT operator which computes a separate discrete density function for each frame of data thereby providing separate histograms for each instant in within a behavioral record.

Often an investigator may wish to explore the temporal structure of behavioral data in greater detail. We have taken several approaches to enable such analyses. Certain simple behaviors may be classified as being merely present or absent and may be denoted by time series of ones and zeros. Several operators (e.g., the LOCAL MAXIMUM, LOCAL MINIMUM, DELTA, INTERVAL DELTA and SPIKE operators) convert real-valued time series into binary valued series. The TIME INTERVAL operator computes the number of frames separating contiguous non-zero elements of such series, thereby allowing the user to determine the temporal intervals which separate discrete events. These time intervals may then be analyzed statistically, resulting in estimates of statistical parameters (using the STATISTICAL PARAMETER operator) and interval histograms (using the SLOT operator).

The analysis of discrete time intervals is a relatively simple procedure inspired by the widespread use of interval histograms among electrophysiologists studying trains of spikes (viz., nerve impulses). We have implemented more complex operators which provide for analysis in the time domain and produce resultant data vectors whose elements are serial correlation (AUTO CORRELATION and CROSS-CORRELATION) coefficients (2). Such operators yield quantitative measures of statistical independence/dependence among neighboring data values within series. They are also helpful in detecting rhythmic or periodic behavior. In addition, we have planned for, but not yet implemented, operators which enable analysis in the frequency domain using Fourier methods (2).

Finally, many behavioral studies entail the investigation of relationships between two or more variables. Figure 7 illustrates the relationship between a simple scalar statistical quantity (viz., sample average linear velocity, a dependent variable) and a real parameter (viz., level of copper exposure, an independent variable). It is possible to study histogrammed distributions in a similar way; however, this requires three-dimensional graphics capability. We have implemented an interactive graphics package (viz., the BRICK operator) which produces fully labeled rectalinear three-dimensional graphs on the plotter. Figures 10 and 11 illustrate the display of separately computed histograms (for direction of travel and rate of change of direction respectively) as functions of copper exposure in *B. improvisus*. This technique can also be used to study the relationship between two behavioral (i.e., inter-dependent) variables such as direction of travel and rate of change of direction (see Figure 12).

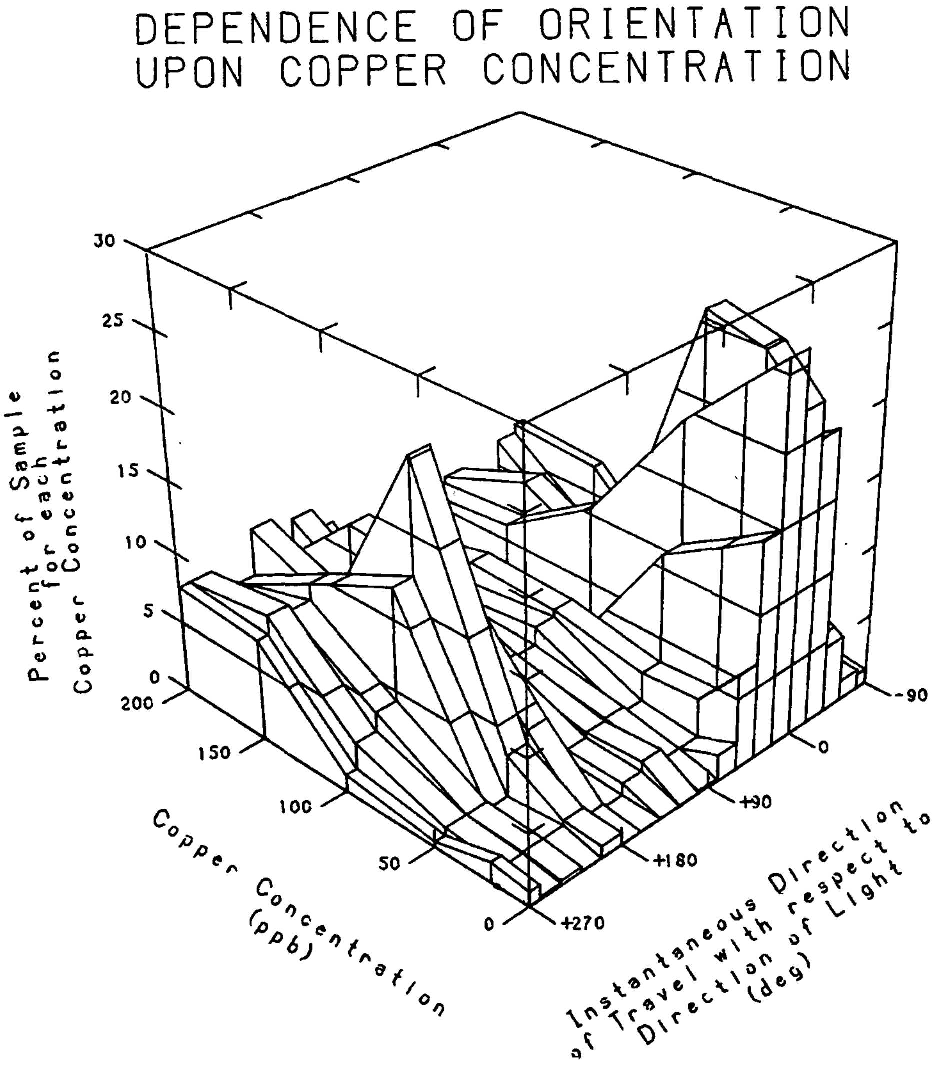


Figure 10. Dependence of phototactic orientation in *B. improvisus* nauplii upon level of copper concentration: the larvae exhibit positive phototaxis at low concentrations of copper (<100 ppb) and negative phototaxis at intermediate levels (100 ppb). No orientation is manifest at high levels of copper exposure (≥150 ppb). Sample sizes range between 521 and 854 data points for each copper concentration. Copper exposure time = 24 hr.

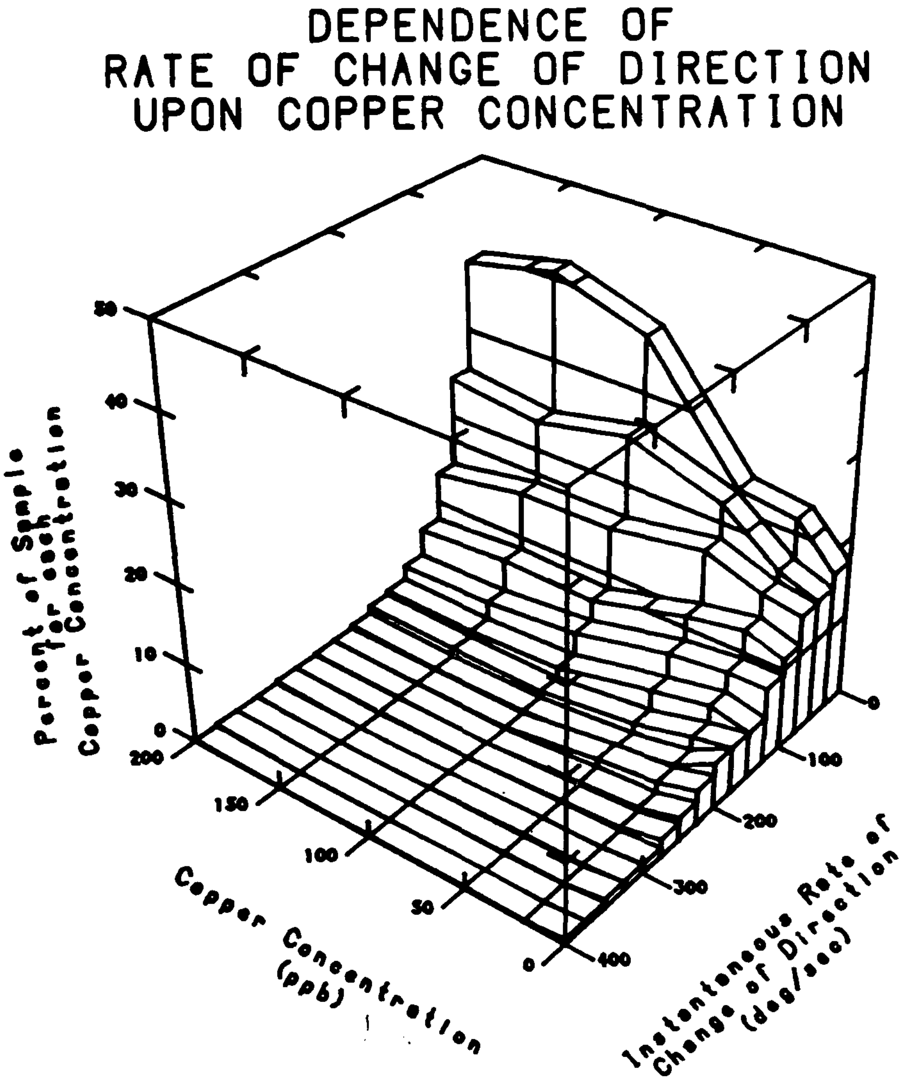


Figure 11. Dependence of rate of change of direction of B. improvisus nauplii upon level of copper exposure. The larvae tend to turn less with increasing concentrations of copper. Sample sizes range between 2055 and 3560 data points. No photic stimulation was present. Copper exposure time = 24 hr.

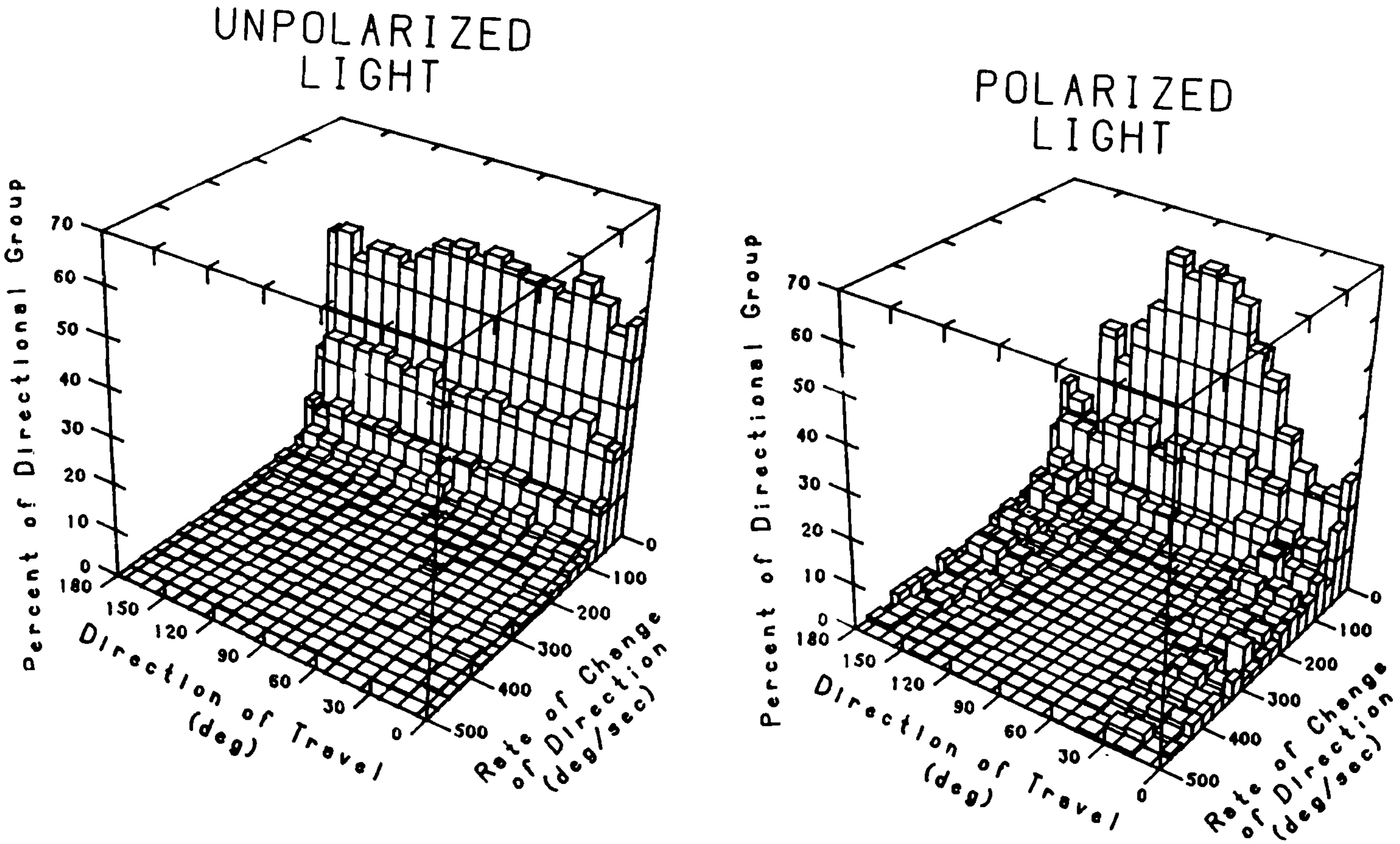


Figure 12. Dependence of rate of change of direction upon direction of travel in *Daphnia pulex* in unpolarized and polarized light. The animals in this experiment orient orthogonal to the e-vector of polarized light (as shown in Figure 4). When they deviate from this preferred orientation thay also turn rapidly, often executing abrupt U-turns.

SECTION IV

DISCUSSION

ORIGINAL PROJECT OBJECTIVES

At the onset of the project, the project goals were to develop a responsive system that would execute on a minicomputer, be as portable as possible, be built from available components using standard television equipment and have one specially built piece of hardware, the Bugwatcher. Also, the system was to be usuable by people with little or no computer background. A discussion of how these goals were met follows.

System responsiveness

When the user of the system sits down to input or analyze data, he or she can collect data at a phenomenal rate through the BUGWATCHER INPUT routine. This alone represents the greatest time savings, with the ability to play back say two minutes of video data and have the Bugwatcher and computer collect and store thousands of data points as the tape is being played back. The software system then responds with a "*" indicating it is ready for the next command. The delay between keyboard operations on the system is usually less than one half second and rarely more than two seconds, which indicates that the system is indeed responsive.

The time required to input and process data is estimated by Wilson as follows. To input and process one hour of video tape, which might typically contain two experimental conditons with twelve replicates each, would take about 15 hours of time on the computer. This analysis includes two sets of experimental conditions (e.g. control and another experimental condition) each having perhaps 10,000 data points of some 200 path trajectories. The processing would include rotating the data points to be aligned with an angular parameter, calculating the respective directions of travel, linear velocities, rates of change of direction, calculating the statistical parameters for the above, listing them on the line printer, generating statistical distributions and plotting these results on the plotter.

Portability

Another design goal was to develop a system that could be exported to other potential users. This objective has been met largely through the choice of FORTRAN as an implementation language. Although there are often "special" features in the FORTRAN language that can be found on one machine but not another, we have resisted temptations to employ them when possible.

The Bugsystem software fully occupies the maximum 32 K-Word memory partition under the RDOS operating system and has several overlays that are swapped in and out at execution time. Such overlay calls are compiler dependent, but are also fairly well isolated in the source files. In addition, there are some features in the software that utilize the 16 bit architecture of the minicomputer; e.g. integers require a single word of storage and real numbers require two such words. The software system was first developed on a PDP11 under the DOS-8 operating system and was made to execute on the same machine under the DOS-9 and, later, the RT11 operating systems. Due to repeated PDP11 hardware failures during the last six months of the project, we stopped supporting the PDP11 software system and put our full effort into the Data General ECLIPSE S/200 under RDOS, now Revision 6.2. To support this software on another PDP11 operating system would take three to four months of programming effort, depending upon programmer experience and familiarity with the Bugsystem software structures. Programming needs might be equally divided into three areas: (1) implementing the driver to communicate with the DMA channel and the Bugwatcher, (2) formulating the appropriate program overlay structures for the new system, and (3) smoothing out differences between the FORTRAN language processors.

Standard Parts

The image collection and storage medium has been standard one-half inch video tape. The RTMA standard 525-line interlaced raster scan cameras have been employed instead of the lower cost random scan cameras, which are not as good in terms of resolution and stability. The best and cleanest image for computer input is recovered from a video deck if the playback unit is manufactured by the same company as the initial video recorder.

The graphics terminal is another standard component. The Tektronix 4010 and 4012 have been used. We did add a special feature to the 4010 terminal on both the ECLIPSE and the PDP11, which we name the "Fastpoint" option. This modification was added to speed up the point displays of video data (which might be 100 pages each with 100 points). The Fastpoint option uses a digital to analog converter on the computer to store a point directly on the face of the storage display screen bypassing the communications procedures. For point displays, this increased the speed of plotting a video data set about an order of magnitude--from about two minutes down to about twelve seconds.

CONCLUSION

In addition to the specific project goals addressed above, a more general objective has guided our development of the Bugsystem: to facilitate highly automated, yet flexible, frame-by-frame analysis of bio-behavioral data. Moreover, we wished to engineer a comprehensive system; i.e., data aquisition, image processing, pathway analysis, statistical analysis and, ultimately, polished tabular and graphical presentation of behavioral data by means of a single system, implemented as an interactive computer language (BRL). We feel that our development efforts have been exceedingly successful in meeting these broader objectives. By means of the Bugsystem, the user is able to collect and analyze behavioral records consisting of tens of thousands of data points, tracking the movement of multiple organisms through space and time. The rich detail inherent to such behavioral data provides *direct* measures of motility; viz., quantitative representations of translational, rotational and flectional movement. Accordingly, we believe that the importance of the Bugsystem is that it makes possible programs of behavioral research which were formerly not physically realizable for "mere" practical reasons. Thus, the modern digital computer promises to offer one route of escape from an old dilemma of behavioral research: the tradeoff between qualitative investigation of biologically significant behavior and quantitative study of activities of dubious biological importance (e.g., interruption of light beams or tripping microswitches). We have little doubt that systems similar to the Bugsystem will become indispensible tools in future investigations of animal behavior.

Development of the Bugsystem at the Environmental Research Laboratory has provided EPA investigators with the technical means to easily record and rapidly quantify the motile behavior of aquatic organisms. Studies are now in progress to determine swimming patterns (both spontaneous and elicited by controlled stimulation) of various marine invertebrate larvae and zooplankton. Modification of behavior in the presence of pollutants is also being compared with results obtained by more traditional assay methods (e.g., LC-50). Significant changes in swimming behavior are manifest when they are subjected to sub-lethal levels of stress (Lang, personal communication). Currently, research is being conducted with the Bugsystem to evaluate the utility of behavioral bioassays to assess water quality and to screen for toxic substances.

SECTION V

REFERENCES

Batschelet, E. 1965. Statistical Methods for the Analysis of Problems in Animal Orientation and Certain Biological Rhythms. AIBS Monograph. Washington, DC. 57 pp.

Chatfield, C. 1975. The Analysis of Time Series: Theory and Practice. Chapman and Hall, London. 263 pp.

Davenport, D., G.J. Culler, J.O.B. Greaves, R.B. Forward and W.G. Hand. 1970. The Investigation of the Behavior of Microorganisms by Computerized Television. IEEE Trans. BME 17: 230-237.

Fraenkel, G.S. and D.L. Gunn. 1961. The Orientation of Animals. Dover. New York. 367 pp.

Greaves, J.O.B. 1971. An On-line Television Computer System for the Study of the Behavior of Microorganisms. PH.D. Dissertation. Dept. Elec. Eng., Univ. of California, Santa Barbara.

Greaves, J.O.B. 1975. The Bugsystem: The Software Structure for the Reduction of Quantized Video Data of Moving Organisms. IEEE Proc. 63: 1415-1425.

Hand, W.G. and J.A. Schmidt. 1975. Phototactic Orientation by the Marine Dinoflagellate Gyrodinium dorsum Kofoid. II. Flagellar Activity and Over-all Response Mechanism, J. Protozool. 22: 494-498.

Martin, W. and K. Brinkman. 1976. A Computer Program System for the Analysis of Equispaced Time Series. J. Interdiscipl. Cycle Res. 7: 251-258.

Wilson, R.S. 1977. Light Elicited Behavior of the Marine Dinoflagellate, Ceratium dens. Ph.D. Dissertation. Dept. Biol. Sci., Univ. of California, Santa Barbara.

Wilson, R.S. and J.O.B. Greaves. 1979. The Evolution of the Bugsystem: Recent Progress in the Analysis of Bio-behavioral Data. In: Proceedings of the symposium on the State of Marine Environmental Research, F. Jacoff, ed. In Press.

TECHNICAL REPORT DATA

(Please read Instructions on the reverse before completing)

1. REPORT NO.
EPA-600/3-80-010

2.

3. RECIPIENT'S ACCESSION NO.

4. TITLE AND SUBTITLE
Development of an interactive system to study sub-lethal effects of pollutants on the behavior of organisms

5. REPORT DATE
January 1980 issuing date

6. PERFORMING ORGANIZATION CODE

7. AUTHOR(S)
John O. B. Greaves and Robert S. Wilson

8. PERFORMING ORGANIZATION REPORT NO.

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Department of Electrical Engineering & Dept. of Biology
Southeastern Massachusetts University
North Dartmouth, Massachusetts 02747

10. PROGRAM ELEMENT NO.
1BA819

11. CONTRACT/GRANT NO.
R803244

12. SPONSORING AGENCY NAME AND ADDRESS
Environmental Research Laboratory - Narragansett, RI
Office of Research and Development
U.S. Environmental Protection Agency
Narragansett, Rhode Island 02882

13. TYPE OF REPORT AND PERIOD COVERED
Final

14. SPONSORING AGENCY CODE
EPA/600/05

15. SUPPLEMENTARY NOTES
---

16. ABSTRACT

Experimental investigation of the movements of organisms often entails the acquisition and processing of large samples of spatio-temporal data. An interactive, interpretive, on-line computer television system (viz., the Bugsystem) was initially developed at the University of California at Santa Barbara in order to expedite such analyses. Aspects of the structure of this prototype system are outlined. Its effectiveness is evaluated with regard to the problems confronting the bio-behavioral researcher.

A second generation system has been developed under a research grant from EPA. Utilizing new hardware and software, it in many ways constitutes a generalization of its prototype. Features of the refined system are described which provide for the following: a large degree of machine-independence, significant expansion of the size of data records, inclusion of experimental parameters and variables within the data structure, investigation of rotational and flectional movement, statistical analysis, advanced graphics and tracking of organisms in three dimentional space. Current utilization of the Bugsystem for future development are discussed.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group
Animal behavior 0510 063 Television systems 1702 Data Processing 0902 Toxicology 0620 Motion 2011 Pollution 1302	Closed circuit television Animal locomotion	57H

18. DISTRIBUTION STATEMENT
RELEASE TO PUBLIC

19. SECURITY CLASS (This Report)
UNCLASSIFIED

20. SECURITY CLASS (This page)
UNCLASSIFIED

21. NO. OF PAGES
45

22. PRICE

☆ U.S. GOVERNMENT PRINTING OFFICE: 1980 -657-146/5536