Behavior Research Methods, Instruments, & Computers
1988, 20 (5), 485-490

INSTRUMENTATION & TECHNIQUES

Monitoring animals' movements using digitized video images

RICHARD F. OLIVO and MARGARET C. THOMPSON
Smith College, Northampton, Massachusetts

An animal's movements can be monitored continuously using video digitization techniques. We outline the differences between video frame-grabbers and column-scan digitizers, and describe two applications using column-scanners: detecting episodes of spontaneous locomotion and tracking the position of a moving appendage. Strategies are discussed for increasing the speed of software and for compressing the information in the video images into an analog motion signal to be displayed and stored for later analysis. Finally, the advantages and limitations of frame-grabbers and column-scan digitizers are assessed.

Many experiments require continuous measurements of an animal's movements. For measuring general ambulatory movements, traditional apparatuses have included running wheels, jiggle boxes, and arenas crossed by infrared light beams (Clarke, Smith, & Justesen, 1985). Measuring the movements of a single appendage has been more difficult, usually requiring the attachment of a device to the appendage. However, the proliferation of inexpensive consumer-grade video products, including video digitizers for personal computers, has made it increasingly feasible to record and measure animals' movements using standard video technology. This paper describes the use of video digitizers, devices that convert video images into numbers in the computer's memory, for two applications: (1) long-term recording of spontaneous locomotion, to detect periods of activity and quiescence, and (2) monitoring of the movements of a single appendage on a tethered animal. Because even a single digitized video image represents a huge amount of data, an experiment that lasts hours demands that the data be compressed substantially before storage or analysis. In both applications, we derived from each digitized video image a single number, which was converted to an analog signal to produce a continuous record of movements. The analog signal was written out, along with other experimental signals, on a chart recorder and instrumentation tape recorder for subsequent analysis.

Margaret C. Thompson was supported by a grant for undergraduate research to Smith College from the Merck Foundation. Her present address is Department of Computer and Information Sciences, University of Massachusetts, Amherst, MA 01003. The manuscript was prepared while Richard F. Olivo was a visiting scientist at the Eaton-Peabody Laboratory, Massachusetts Eye & Ear Infirmary, Boston; he thanks the members of the laboratory and its director, Nelson Kiang, for hospitality. We are grateful to Carrie Fischer, Dennis Freeman, and Ishmael Stefanov-Wagner for their comments on the manuscript. Address correspondence to Richard Olivo, Department of Biological Sciences, Smith College, Northampton, MA 01063.

Video Digitizers

Two types of video digitizers are available for microcomputers: frame-grabbers and column-scan digitizers. To understand their differences, it is helpful to review a few basics of video technology.

A standard video signal from a camera or video recorder is an analog electrical signal that controls the brightness of points on a video monitor's screen. To paint a video image, an electron beam that is modulated by the analog signal sweeps across the screen in a series of horizontal scan lines. Synchronizing pulses embedded in the signal bring the electron beam to the top of the screen (vertical sync) and the left edge (horizontal sync). The full screen (a video "frame") is repainted 30 times each second.

Each video frame consists of 525 horizontal scan lines, of which 480 are visible. Since 30 frames per second would create a disturbing flicker, each frame is painted as two interlaced "fields," with the even scan lines in one field and the odd lines in the other, for a total of 60 fields per second. Each scan line lasts only 64 μsec, a very brief interval even by computer standards. When an image is digitized, each scan line is typically divided into at least 256 (and often 512 or more) discrete points, called picture elements or pixels. Thus, every 64 μsec, a video digitizer must convert the analog signal into several hundred stored numbers, representing the gray values of pixels along the scan line. This task is much too fast for ordinary microprocessors and analog-to-digital (A/D) converters.

Two solutions have emerged for coping with the high speed of video signals. The first, used in frame-grabber video digitizers, is to employ a very fast, relatively expensive flash converter. The devices also have on-board high-speed memory to hold the video image, and associated circuitry to reconvert the stored digitized image to a displayable video signal. The high speed of the converter and the memory permit the capture and storage of complete video frames in real time. Frame-grabbers for black-and-white video with a resolution of 480 (vertical) × 512 or 640 (horizontal) are currently available for the PC/AT bus at prices starting at $1500, and for the Macintosh II at $1,000.

The second solution, column-scan video digitization, is less expensive and is available in devices for low-end computers such as the Commodore 64 and Apple II (see Appendix for manufacturers and addresses). Column-scan digitizers use moderate-speed A/D converters, and usually place their samples in the computer's own graphics memory. Since the moderate speed with which these devices sample and store pixels cannot keep up with the video signal, a different approach is taken: only a single pixel is digitized for each horizontal scan line. Under software control, the computer's processor provides the digitizer with the horizontal and vertical coordinates of the next point to be digitized. When the coordinates of the incoming video image match the requested coordinates, a sample-and-hold circuit captures the analog video voltage. The captured signal is then converted to the digital gray value for that pixel, after which the processor transfers the value to an appropriate address in the computer's memory. To acquire a full image, software directs the systematic sampling of points. Typically, beginning at the left edge, one point is sampled from each horizontal scan line; at the end of one video field, a vertical column of points has been acquired. The software then specifies a horizontal position one pixel to the right, and during the next video field, another vertical column of points is acquired. As one watches the image being built on the computer screen, it appears that an active column is moving across the screen. Scanning a full image of 256 columns requires 256 video fields; at 16.7 msec per field, acquiring a full image takes 4.3 sec.

The technical differences between frame-grabbers and column-scan digitizers give each device advantages and limitations. The most obvious difference is that column-scan digitizers do not have the fast temporal resolution of frame-grabbers, and thus they cannot be used for acquiring accurate images of rapidly moving animals. Also, column-scanners (and some lower cost frame-grabbers) generally do not distinguish between odd and even video fields. Instead, they treat pairs of horizontal lines as if they were the same, and consequently they have lower spatial resolution than most frame-grabbers. Nevertheless, the low cost of column-scanners, and other factors that we discuss later, make them attractive for some applications.

We describe next two examples in which a column-scan digitizer was used to monitor movements. The first example demonstrates the use of a digitizer to detect active and quiescent levels of locomotion; the second shows how, by restricting the measurement zone to a single vertical column, rapid movements of an edge can be tracked accurately in real time.

Monitoring Episodes of Spontaneous Locomotion

Our first example that demonstrates how video digitizers can be used is taken from a series of experiments in which the spontaneous walking movements of tethered crayfish were monitored for many consecutive hours (Arnesen & Olivo, in press; Olivo & Thompson, 1982). We mounted a black-and-white video camera directly above an animal, which walked on a floating foam-rubber ball. In the video image, the dark crayfish was easily distinguished from the light-colored objects that formed the background. The image was digitized repeatedly, and inspected for changes that indicated movement.

The strategy for detecting movement was as follows. A rectangular scan area was established to include the crayfish and its appendages, but to exclude as much of the remaining background as possible. Each pixel in the scan area was compressed to a single bit in memory (1 or 0) according to whether its gray value was darker or brighter than a threshold gray (which we chose in advance to separate the animal from its background). The compressed image was compared pixel-for-pixel with the previous image, and all pixels that differed in the two images were counted. This count indicated the number of pixels that had been "crayfish" in one image and "background" in the next, or vice versa; it provided a measure of the extent to which movement had occurred. To display and record this information, the count was then scaled to a range of 0 through 255, and was sent to an 8-bit D/A converter to produce a voltage signal proportional to the number of pixels that had changed between scans. The output voltage, updated after each scan, gave us a continuous record of the animal's movement.

Since the information about each image was not retained once a count had been made, it was important to be able to monitor the process visually to guarantee that the reduced data were free of artifacts. The software allowed us to display either the compressed image or an image that showed only the pixels that had changed, referred to as a *difference scan*. Figure 1 illustrates these two types of displayed images. Figure 1a is a compressed image, in which each pixel is either black or white. Individual pixels (dots) are readily resolved, indicating the relatively crude spatial resolution (here, a 128×128 array) that is adequate for motion detection. The remaining parts of Figure 1 show difference scans. In Figure 1b, the crayfish was motionless. A fringe of scattered pixels appears around its edges, representing noise in the measurement. These pixels are probably due in part to electronic noise in the video signal, which puts some marginal pixels above the gray threshold in one image and below the threshold in another. A second source of the edge pixels is the digitizer's inability to distinguish between odd and even video fields, so that pairs of pixels on adjacent video lines are treated as if they were the same. In any case, the noise pixels introduce a relatively constant baseline in the motion signal. When the animal moved, however, the count

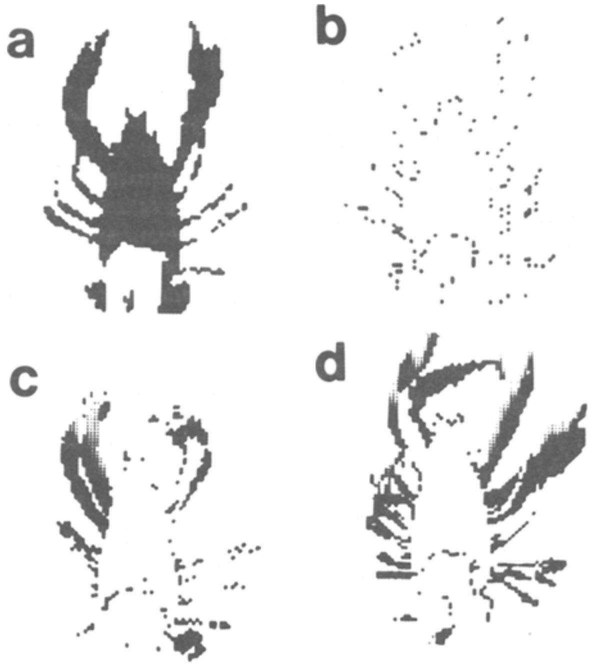

Figure 1. Digitized video images of a tethered crayfish, viewed from above. The animal is held in place by a plastic rod glued to its back; its legs hold a floating foam-rubber ball, and are free to move. (a) Compressed image, showing all points darker than the threshold gray level. The white region at the bottom of the dark crayfish is the light-colored support rod that held the animal in place. (b) Difference scan, showing all points that changed from white to black, or black to white, between the most recent and the previous image. The animal was sitting quietly, without moving; the scattered dark pixels represent noise, as described in the text. (c) Difference scan during slight movement, particularly of the left claw, which shows its advancing and retreating edges. (Variations in dot size are an artifact of the oscilloscope's beam-brightening circuitry.) (d) Difference scan during pronounced movement of the claws and walking legs. Most of the appendages appear twice because the pixels that represent them have changed between images.

of changed pixels and the analog signal increased dramatically above the baseline. In Figure 1c, slight motion occurred. (Note the large increase in the total number of pixels that appear, including partial double images of each claw. A moving appendage appears as a double image, because pixels have changed at both its original position and its current position: from animal to background where the appendage was, and from background to animal where it now is.) In Figure 1d, greater motion occurred, with both claws and most of the walking legs changing positions.

Since the resolution of the original compressed image is relatively low, and since the difference scan represents data acquired over many frames, and thus would differ slightly if the starting frame were earlier or later, we were curious about the reproducibility of the analog motion signal. Figure 2 illustrates the outcome of one test. Each of the figure's four parts shows the scan area and the corresponding analog motion signal for the same few minutes of behavior. In this sequence, three prolonged episodes of movement were separated by brief quiescent periods. In Figure 2a, the measurement was made on the live video signal; in Figure 2b, the same scan area was measured on playback of a videotape, with the starting frame for the scan slightly displaced in time. In comparing these two motion records, it is clear that the timing and rough shape of the records are similar, but the fine structure of the analog signals is not the same. On additional playbacks, with the scan area limited to the claws alone (Figure 2c) or the legs alone (Figure 2d), the basic episodes of movement again appeared equivalent, but the details of the records were different. We conclude from such experiments that the analog motion records should be treated as rough but not detailed indicators of the timing and extent of the animal's movements. Nevertheless, the records are entirely reliable for distinguishing periods of

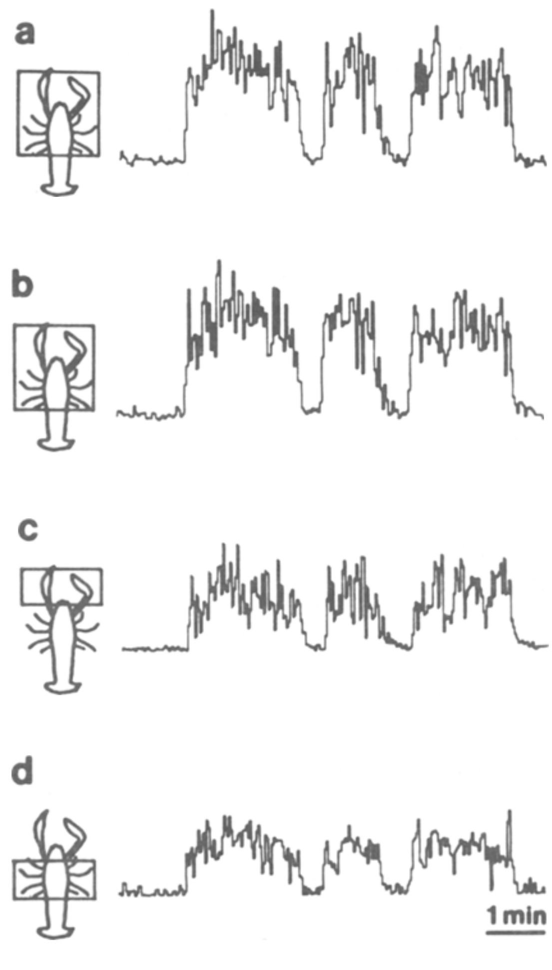

Figure 2. Reproducibility of video-based motion detection. Each trace shows an analog signal proportional to the number of pixels that have changed between scans. (a) Video scan covering the claws and walking legs, measured from the live video signal. (b) A scan of the same area and the same behavior, but measured on playback of a videotape recording. (c) A scan of the claws only, measured from the videotape. (d) A scan of the walking legs only, measured from the videotape. (In this trace, the vertical amplification is twice that of the other traces.) Note that although the fine structure of the analog signal is different for each record, the three sequential episodes of movement are clearly detectable in all records and are similar in their general timing and shape.

quiescence from periods of spontaneous behavioral arousal.

Whether a frame-grabber or column-scan digitizer is used, the software must be efficient because the image contains a huge amount of data (up to 256K for a $512 \times 512 \times 8$-bit frame) and the motion signal must be updated in real time. In addition, for a column-scan digitizer, it is essential that the software be fast enough to specify x–y coordinates and to store pixel gray values for each scan line, without exceptions. Otherwise, the digitizer would have to wait an entire field (16.7 msec) instead of one scan line (64 μsec) between points, which would drastically degrade the imaging rate. Using a 6502 microprocessor at 1 MHz and programming directly in assembly language, we found it fastest to place raw gray values from the column-scan in a temporary array, using a processor register to index the array and to specify the vertical coordinate. Once an entire column of points had been stored, the software had time to make additional calculations for the column of pixels. It compressed the image by comparing each value in the temporary array to the gray threshold, and packed the resulting bits into bytes. Each packed byte was then exclusive-ORed with the equivalent byte from the previous compressed image to produce the difference byte, representing the pixels that had changed. The difference byte and the compressed image byte were each stored in their respective image arrays, so that both arrays (the compressed and difference images) were gradually transformed from their old to their new values as the column-scan progressed. (If a frame-grabber were used, it would be necessary to store two complete compressed images before calculating the differences between them, although the strategy for comparing pixels would be the same as with a column-scan digitizer.) Another improvement that made the final software run faster was to create a 256-entry table giving the number of 1s in every 8-bit number. Bytes from the difference scan served as an index into the table, thereby calculating the number of changed pixels with a single instruction rather than having to shift each difference byte eight times to count the 1-bits. Finally, by making the scanned area cover less than the entire image, and by using every other column and line (a 128×128 array instead of 256×256), we made the process run faster (one image every 2.2 sec) while still obtaining adequate spatial resolution.

Monitoring the Position of an Appendage

A second type of motion monitoring is to track the position of a dark-light boundary, such as the edge of a moving appendage. As a test of our method, we analyzed videotaped sequences supplied by Thomas Consi and Eduardo Macagno of Columbia University, who recorded reflex flicks of a tiny crustacean's eye in response to flashes of light. In their experiments, the animal, *Daphnia*, was exposed to flashes of various wavelengths and intensities to obtain behavioral data on its spectral sensitivity (Consi & Macagno, 1985). The animals were videotaped through a microscope in infrared illumination; for our analysis, they were oriented so that their eye flicks occurred in a vertical direction. By tracking the position of the eye, the software produced a continuous record of eye flicks, making it possible to monitor which flashes were above threshold. In this animal, the eye's small size (about 200 μ) and location beneath the (transparent) body surface make it impossible to attach a measuring device to the moving structure; consequently, the example is an excellent demonstration of the advantages of video techniques, which do not require direct contact.

The strategy for tracking the eye's movements was to place a vertical scan-column 1 pixel wide across the lower edge of the eye, as shown in Figure 3a. When the eye flicked, it rotated in a way that moved its lower edge up and down in the scan zone. To detect the position of the edge, the software placed the gray values from the scan zone in a temporary array, as in the previous example, and then inspected the array to find the lowest pixel in the column that was darker than a preset (and relatively dark) threshold. This separated the very dark eye from some of the other background structures. Figure 3b shows x–y plots of gray values for pixels in the scan zone, with plots for two orientations of the eye superimposed. Arrows indicate the positions of the eye's edge as detected by the software. After each vertical scan, the detected vertical coordinate of the edge was converted to an analog voltage (Figure 3c) to provide a continuous signal of the eye's position. The signal was updated every 16.7 msec, or once for each video field, which is the maximum rate that can be obtained from a standard video signal. Although for optimal speed the scan zone must be vertical, and for our software it must not exceed 160 horizontal lines (the remaining time in each video field is required for calculations), this experiment shows the feasibility of using column-scan digitizers for measurements with high temporal resolution. In fact, because of some of the limitations discussed in the next section, frame-grabbers might not be able to make measurements as quickly.

We have not discussed the hardware details of our implementation until this point because we used a now-outmoded Rockwell AIM-65 microcomputer (Heth, 1980) that would not be chosen today. (A Commodore 64 or an 8088-based PC clone would probably be our choice for a very low-cost system, and a Macintosh II or a PC/AT clone with a frame-grabber for a higher cost system.) We modified a column-scan digitizer (Digisector DS-68, The Micro Works, Inc.) to communicate with two parallel ports on the AIM-65, and created analog outputs for the motion signal and for an x–y display on an oscilloscope (for graphics, which the AIM-65 lacks) by attaching three 8-bit D/A converters (AD558, Analog Devices, Inc.) to parallel ports on an accessory card. For a personal computer with its own graphics, only a single analog output for the motion signal would have to be added. A wide variety of video digitizers now is available from many manufacturers (see the Appendix, and also Linzmayer, 1985), as are analog input/output cards; no hard-

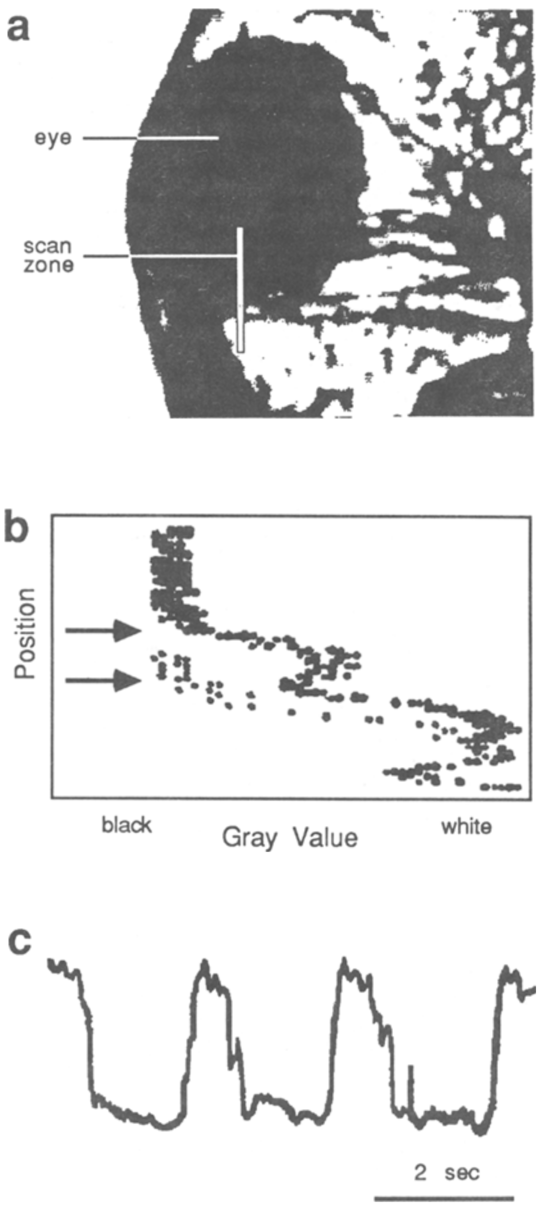

Figure 3. Tracking a moving edge in real time using a narrow vertical scan zone. (a) High-contrast image of the eye of a miniature aquatic crustacean, *Daphnia*, as seen from the side. (The original video image, obtained through a microscope in infrared light, had a full gray scale, but was converted to high contrast for reproduction here.) A vertical scan zone 50 pixels high and 1 pixel wide was positioned across the bottom edge of the eye, to detect up-and-down movements. (b) Gray values of pixels in the scan zone, plotted for two positions of the eye. Each dot represents the gray value of a single pixel. The axes are arranged so that the vertical axis, position in the scan zone, has the same orientation as the vertical scan zone in the image above. Most of the pixels are from several sequential video frames taken when the eye was at rest. Also plotted are pixels from the lower part of the scan zone for a single video frame taken during an eye flick. Arrows mark the detected position of the edge (the point where gray values shift from black toward white) for the eye's two different positions. (c) Excerpts from an analog chart record of the eye's position, showing three eye flicks (downward deflections of the trace).

ware modification is required to use them. However, most of the digitizers and their software are designed for capturing images for printing and display, and not for making measurements. As a result, detailed programming will usually be necessary for applications of the type described here.

Column-Scanners Versus Frame-Grabbers

The two applications we have presented, generalized motion detection and edge tracking, are examples of a wide range of possible video-based measurements. The basic strategy we used for detecting spontaneous locomotion could also be applied, for example, to monitoring the running of unrestrained rodents in an arena. Similarly, edge tracking could track leg movments or the position of the pupil of an eye. For any particular application, the basic software strategy for extracting data from a video image is the same whether a frame-grabber or a column-scan digitizer is used, aside from the details of how the image is acquired. Although they have limitations, low-cost column-scan digitizers can be used successfully in some applications, and may even be preferable for edge tracking. However, as frame-grabbers become cheaper, their versatility will increasingly make them the optimal choice. They have the obvious advantage of acquiring the image instantaneously, eliminating motion distortion, and freeing the overall process from the timing limitations of column-scanning. Frame-grabbers also typically provide enough memory to store multiple images; they have input look-up tables that permit automatic compression of an image during its acquisition; and they have advanced output features, such as pseudocoloring, that make it possible to display differently colored overlays of the compressed image and the difference scan.

However, some of our recent attempts to track unrestrained animals with a PC/AT-based frame-grabber (Data Translation DT-2851; Olivo and Pufall, unpublished experiments) indicate that there also are disadvantages to using frame-grabbers. The main problem stems from the large memory space that is needed to hold a high-resolution image. The (nominal) $512 \times 512 \times 8$-bit image from a typical frame-grabber occupies 256K of high-speed memory, but the PC/AT architecture usually requires that this image memory be located above the 1-MB memory space that MS-DOS can address. As a consequence, currently available languages cannot address pixels directly; instead, all or part of each captured frame must be copied down into the lower 640K of memory before manipulation or inspection of the image can be made by software. (Details of how to do this in the Forth language are available from the authors.) In addition, once the image has been copied, even a fast processor requires time to inspect 256,000 pixels. This means that although frames can be acquired as fast as the camera generates them, they cannot be processed in real time. One must either skip a certain number of frames between samples, or work with frame-by-frame playback from a video recorder. As a result, the position of a freely moving animal cannot be tracked dependably in real time through digital inspection of the image. Instead, it is necessary to use analog techniques that return the image coordinates when the video signal exceeds a preset voltage level (Crawley, Szara, Pryor, Creveling, & Bernard, 1982; Fleischer & Pflugradt, 1977). Even so, frame-grabbers are, at present, the best choice for general monitoring of motion, and their advantages will increase. New operating systems will remove MS-DOS's limitations on directly addressing image memory; attractively priced frame-grabbers are now appearing for the Macintosh II, which does not have the crippled address space of MS-DOS computers; and, for the type of applications that we have discussed here, software can be made to run faster by not sampling every pixel.

As indicated, video techniques have become attractive and cost-effective for measuring live movements. When extended to frame-by-frame playback, where software speed is not as critical, digitized video can also be used for more complex analyses of behavior. For example, items from a repertoire of behavioral fragments can be identified by matching an animal's posture to a series of stereotyped templates (Kernan et al., 1980). Eventually, image-recognition techniques, now much too slow for practical use, may bring the benefits of artificial intelligence to the automated analysis of behavior. In the meantime, by using the simpler techniques that we have described, it is entirely feasible to extract continuous measures of movement in real time from digitized video images.

REFERENCES

Arnesen, S. J., & Olivo, R. F. (in press). The effects of serotonin and octopamine on behavioral arousal in the crayfish. *Comparative Biochemistry & Physiology.*

Clarke, R. L., Smith, R. F., & Justesen, D. R. (1985). An infrared device for detecting locomotor activity. *Behavior Research Methods, Instruments, & Computers*, **17**, 519-525.

Consi, T. R., & Macagno, E. R. (1985). The spectral sensitivity of eye-movements in response to light-flashes in *Daphnia magna*. *Journal of Comparative Physiology*, **156A**, 135-143.

Crawley, J. N., Szara, S., Pryor, G. T., Creveling, C. R., & Bernard, B. K. (1982). Development and evaluation of a computer-automated color TV tracking system for automatic recording of the social and exploratory behavior of small animals. *Journal of Neuroscience Methods*, **5**, 235-247.

Fleischer, A., & Pflugradt, M. (1977). Continuous registration of X, Y-coordinates and angular position in behavioural experiments. *Experientia*, **33**, 693-695.

Heth, C. D. (1980). The AIM-65 microcomputer as a laboratory control device. *Behavior Research Methods & Instrumentation*, **12**, 364-366.

Kernan, W. J., Jr., Higby, W. J., Hopper, D. L., Cunningham, W., Lloyd, W. E., & Reiter, L. (1980). Pattern recognition of behavioral events in the nonhuman primate. *Behavior Research Methods & Instrumentation*, **12**, 524-534.

Linzmayer, O. (1985). Digital image processors. *Creative Computing*, **11(7)**, 65-73.

Olivo, R. F., & Thompson, M. C. (1982). Behavioral arousal in the crayfish produces a rapid increase in optokinetic gain. *Society for Neuroscience Abstracts*, **8**, 735.

APPENDIX
Some Manufacturers of Video Digitizers for Personal Computers

ATD/Zuckerboard
235 Santa Ana Court
Sunnyvale, CA 94086
(408) 720-1942
IBM PS/2

Chorus Data Systems, Inc.
P.O. Box 370
6 Continental Boulevard
Merrimack, NH 03054
(603) 424-2900
IBM PC

Data Translation, Inc.
100 Locke Drive
Marlboro, MA 01752
(617) 481-3700
IBM PC, PC/AT, Macintosh II, MicroVAX

DATACUBE, Inc.
4 Dearborn Road
Peabody, MA 01960
(617) 535-6644
IBM PC, PC/AT

Digital Vision, Inc.
66 Eastern Avenue
Dedham, MA 02026
(800) 346-0090
Commodore 64, Apple II, IBM PC, Atari ST

Imaging Technology, Inc.
600 West Cummings Park
Woburn, MA 01801
(617) 938-8444
IBM PC, PC/AT, VMEbus, Q-bus, Multibus

Koala Technologies
269 Mount Hermon Road
Scotts Valley, CA 95066
(408) 438-0946
Macintosh

Matrox Electronic Systems Ltd.
1055 St. Regis Boulevard
Dorval, Quebec H9P 2T4, Canada
(514) 685-2630
IBM PC, PC/AT, VMEbus, Q-bus

Micro Works
P.O. Box 1110
Del Mar, CA 92014
(619) 942-2400
Apple II, IBM PC, S-100, S-50

Pixelogic, Inc.
38 Montvale Avenue
Stoneham, MA 02180
(617) 438-5520
Macintosh

Truevision, Inc.
7351 Shadeland Station, Suite 100
Indianapolis, IN 46256
(800) 858-8783
IBM PC, PC/AT, Macintosh II

(Manuscript received February 23, 1988;
revision accepted for publication June 7, 1988.)