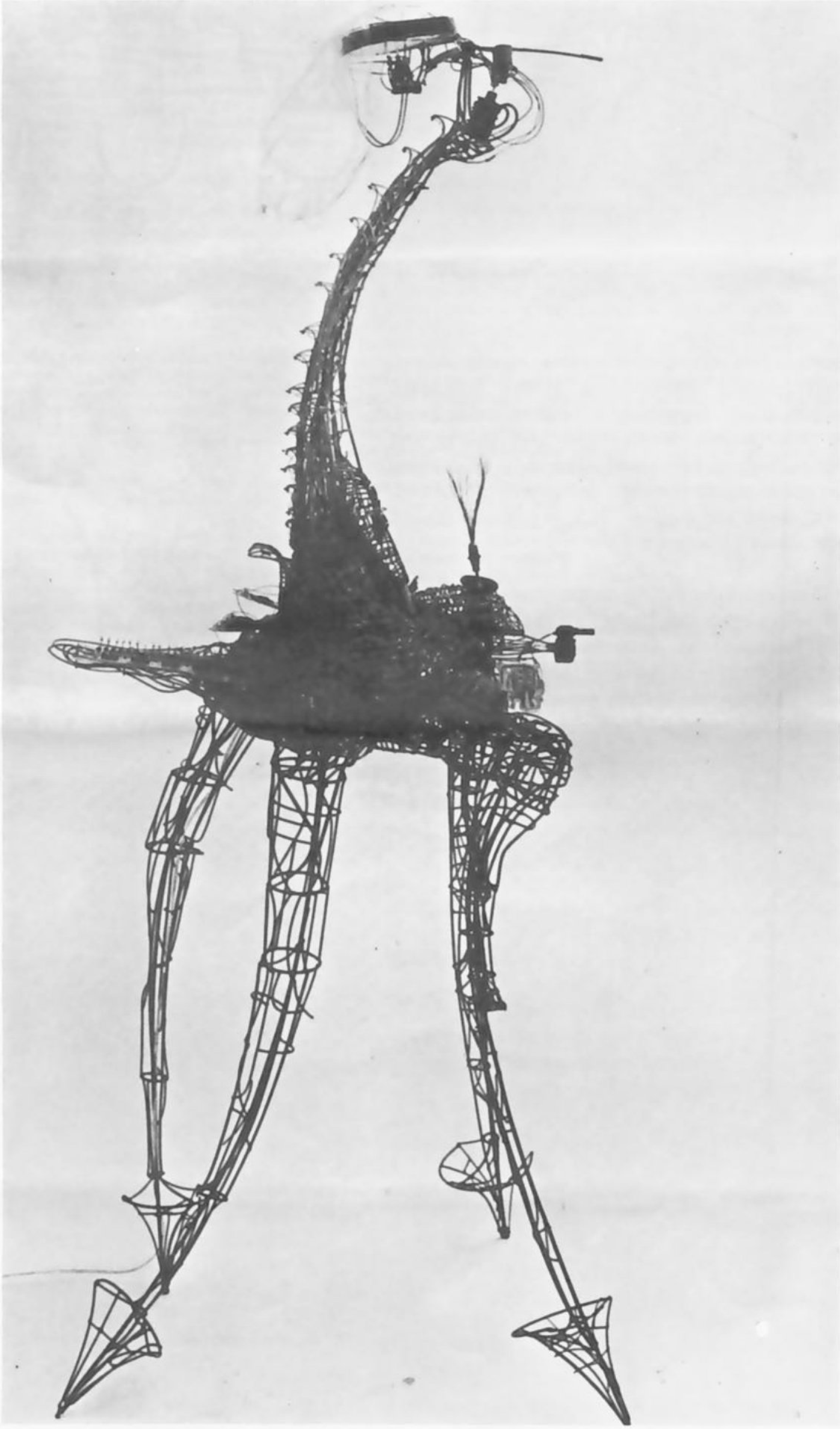
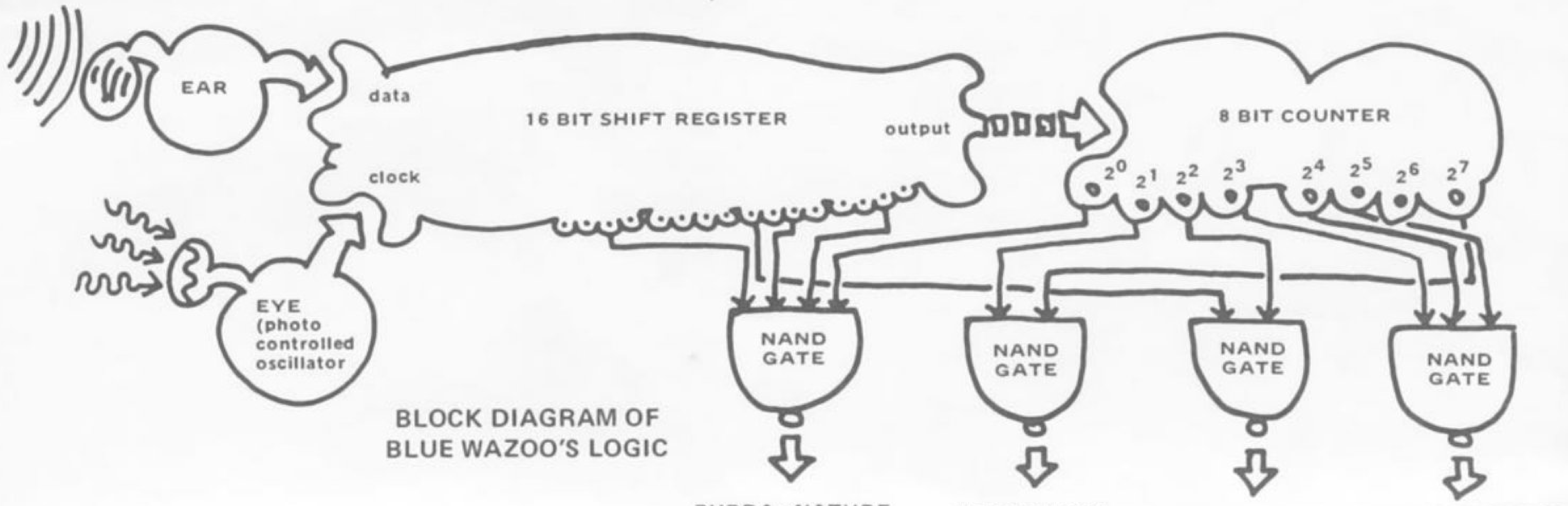
Page 39
BULLETIN OF THE COMPUTER ARTS SOCIETY
NOVEMBER, 1977





BLOCK DIAGRAM OF
BLUE WAZOO'S LOGIC

The Blue Wazoo is primarily a welded steel structure covered with several coats of acrylic lacquer. The structure contains plastic shapes, circuitry, wires, light-emitting diodes, solenoids, a motor, cloth, horsehair, a feather and a bead. The Blue Wazoo senses light and sound and responds with a behavioral repertoire of various LED patterns, movements, inflations, deflations, whirrs, clicks and jiggles. It is six feet high and weighs about 25 pounds.

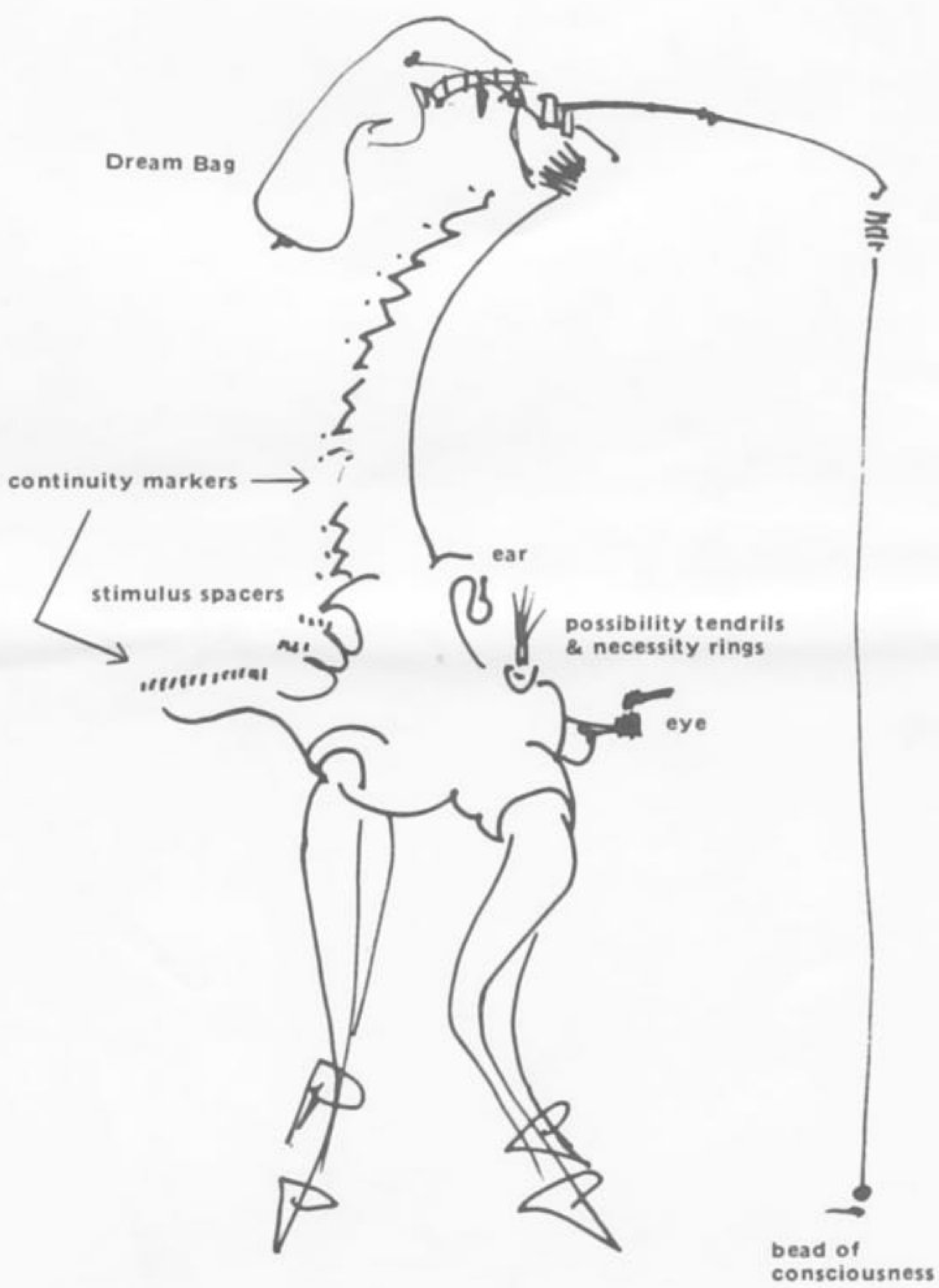
Ambient light falling on the photo cell (in the small velcro-mounted "gun" at the front of the body) increases the clock frequency (average; 2 hertz), of the 16 bit serial-in, serial-out, shift register. Data for the shift is generated by a microphone (at the front base of the neck). An 8 bit binary counter counts the output of the shift. Four Nand gates are made conditional on data from various outputs of the shift and counters. Each Nand gate controls an activity of the Blue Wazoo.

Visitors sense that the Blue Wazoo is reacting to something they are doing but they don't know exactly what. This often leads to superstitious behavior on their part. The artist is particularly interested in this cybernetic aspect of the work and feels that the dynamic interaction between viewer and artwork is one of the more exciting potentials in the use of the new technology in art.

The internal workings of a circuit and the resultant activities take on symbolic content as indicated by the individual names given to various parts of the sculpture.

The Blue Wazoo was made by Jim Pallas in 1975-76 and is in the collection of the Allan Stone Gallery, NYC.

Jim Pallas is co-ordinator of the Detroit Art Works and on the faculty of Macomb County Community College.



BLUE WAZOO

THE BEAD OF CONSCIOUSNESS SUSPENDS
CONTINUITY MARKING,
SPACED STIMULI.
BUDDHA-NATURE WAVERS TWITCH.
THE DREAM BAG FILLS.
POSSIBILITY TENDRILS
AND NECCESSITY RINGS
POP UP.

BEAD DESCENDS.

Computer-Assisted Composition and Performance Control of Theatrical Lighting and Effects.

M. Garrard, E.R. Ghent, W. Hemsath, J. Seawright

Under the sponsorship of the Mimi Garrard Dance Company, and with the support of a Special Project Grant from the National Endowment for the Arts, a comprehensive study and prototype development has been completed leading to the construction of a computer-assisted composition and performance control unit for theatrical lighting and effects. This research was based on work by M.V. Mathews and F.R. Moore, and others, of Bell Telephone Laboratories, Murray Hill, New Jersey, and on previous work by the authors.

The new system prototype is constructed of inexpensive and widely available microcomputer components and may be configured in a variety of ways to suite special needs. We have adopted a standard configuration with the following characteristics and specifications:

System units

1. A console unit which consists of a microprocessor with disc memory, an ASCII keyboard, a video monitor display, and other input and output devices for handling performance data.
2. A synchronizing system, which allows the composed lighting control data to be synchronized with recorded sound by magnetic digital recording on another track of the sound tape.
3. A dimmer unit, which translates the lighting control data into brightness levels of standard theatrical lighting devices.

System specifications

1. Control of up to 32 channels of information, each channel controlling up to 2500 watts of power (channels may be paralleled).
2. Brightness control of up to 256 discrete levels.
3. Production of any effect which can be defined in 1/10 second samples (the practical response time of the lighting instruments) on all 32 channels simultaneously.
4. Easy, direct compositional and editing control of over 13 minutes of duration of all 32 channels.
5. Unlimited compositional duration (in segments of up to 13 + minutes)
6. Exact, repetable performance in perfect synchronization with sound or other cueing.

System Description

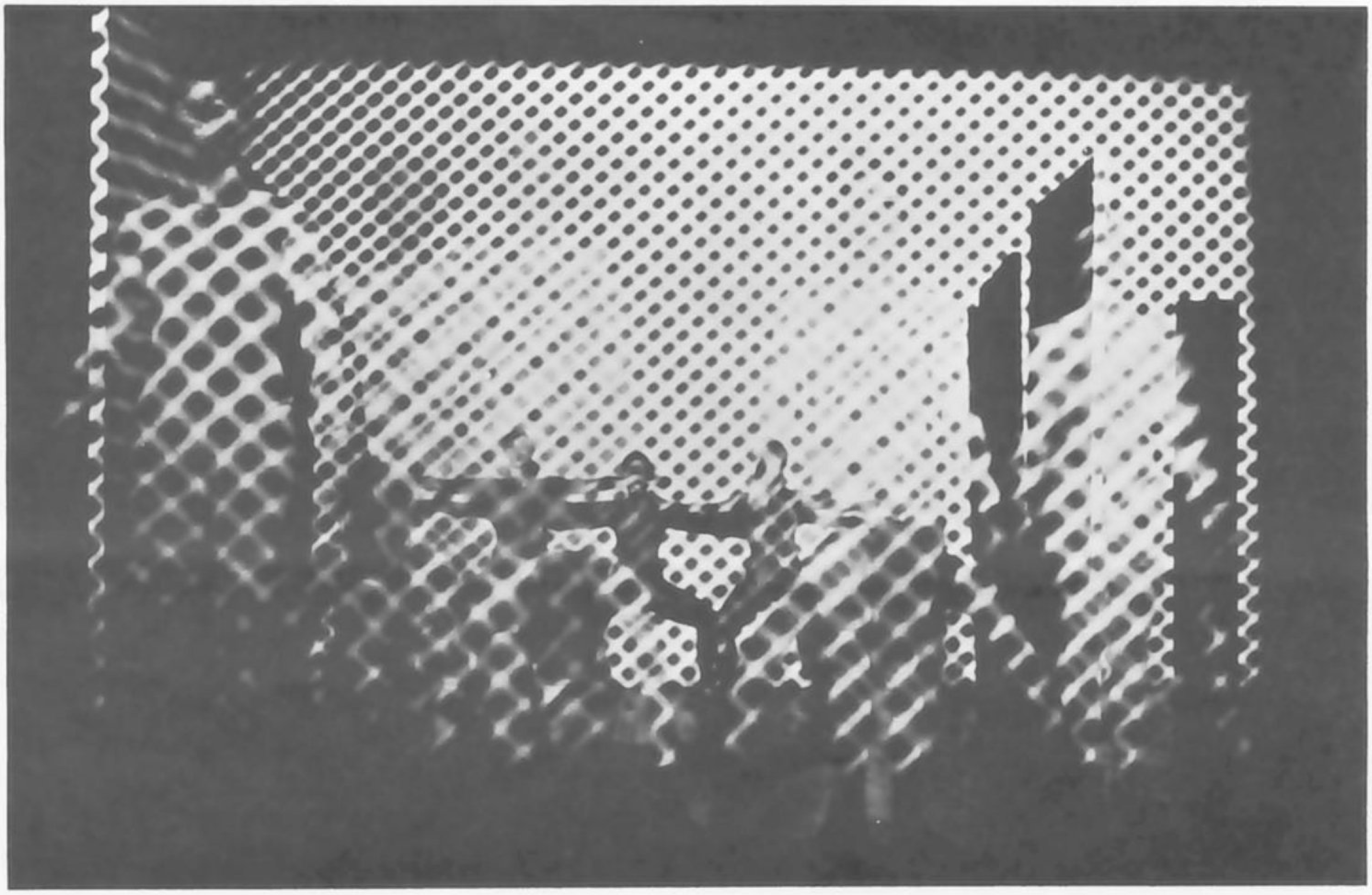
a. Hardware (see diagrams 1 and 2)

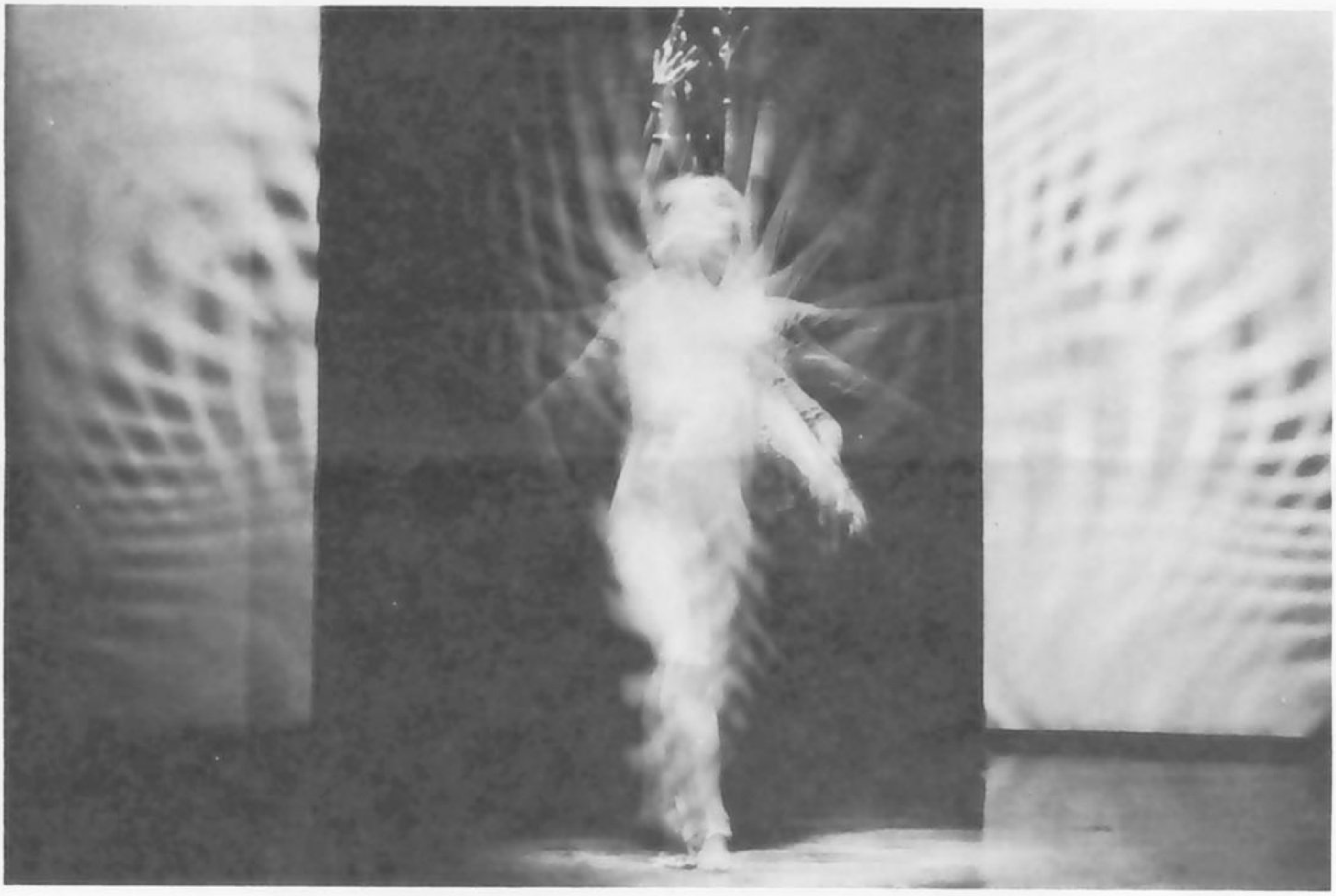
We have chosen to use elements of the 8080 microprocessor technology, as this seems to bring the widest range of commerically-available components and accessories and the most extensive software support at present. Our system consists of a console unit cabinet containing a power supply and a 5-100 bus, into which the various circuit cards may be plugged. (The 5-100 bus is an industry-standard arrangement of 100-pin sockets connected together with all similar numbered pins common — each circuit card is in effect connected to all of the others through the appropriate pins of the common 5-100 bus).

In the cabinet are circuit cards for:

1. 8080 microprocessor
2. random-access memory
3. read-only memory
4. brightness control processor
5. video display control
6. disc memory control
7. input-output control (keyboard, tape in out)
8. analog/digital and digital/analog conversion

All of the above are commerically available except for (4) brightness control processor, although we have modified others to suit the purposes of the system.





The remaining parts of the console unit are the disc memory itself (a "Floppy" disc — IBM style 7" flexible disc drive) a 9" video monitor, and a standard ASCII keyboard with a special knob control for composing time functions.

The synchronizing unit consists of an electronic digital recording control and accessory record/play head assembly which can be attached to a variety of commerical audio tape recorders and which allow recording and playback of light control data on the same tape on which the sound track is recorded.

The dimmer unit consists of modular 8-channel dimmers which regulate the flow of AC power to lights in response to the control signals from the console unit.

b. Software

The system program is designed to suit the configuration of the prototype. However, a wide range of support programs have been developed which make possible revision and rearrangement of the basic composing program or the development of new programs altogether. These programs include the disc operating system, peripheral interchange, text editing, assembly, format conversion, general purpose computation, test and diagnostic programs. These will be described in the full report.

Basic composing program: (see diagram 3)

The basic composing program can best be understood by consideration of the video display presented to the operator and the operational command structure of the program.

Initially, the lighting designer has to consider how to dispose the resources of the lighting instruments available. They must be hung around and about the space to be lighted, and focused, and gelled to the desired color. If image-projection devices are to be used they must be placed and the images selected and ordered in sequence. The lighting designer will then have a plot of lights identified by number and assigned to channels 0 through 31.

At the console the display diagrammed in diagram 3 is seen. Six graphical "score" lines are seen which may be assigned to any six of the 32 (0 to 31) channels. In the upper left will appear input from the keyboard, error or output messages. In the upper right is seen the current time indication and the value of the input knob (if in use). Below the six score graphs are absolute time marks. Any display shows a 5-second "page" of a total time span of 770 seconds. Current time is indicated by the position of a vertical bar called the "cursor." The cursor's position in absolute time units is also shown at the upper right as noted above. At the bottom of the display is space where text may indicate synchronizing marks in reference to the sound score.

DIAGRAM 1
BLOCK DIAGRAM OF SYSTEM

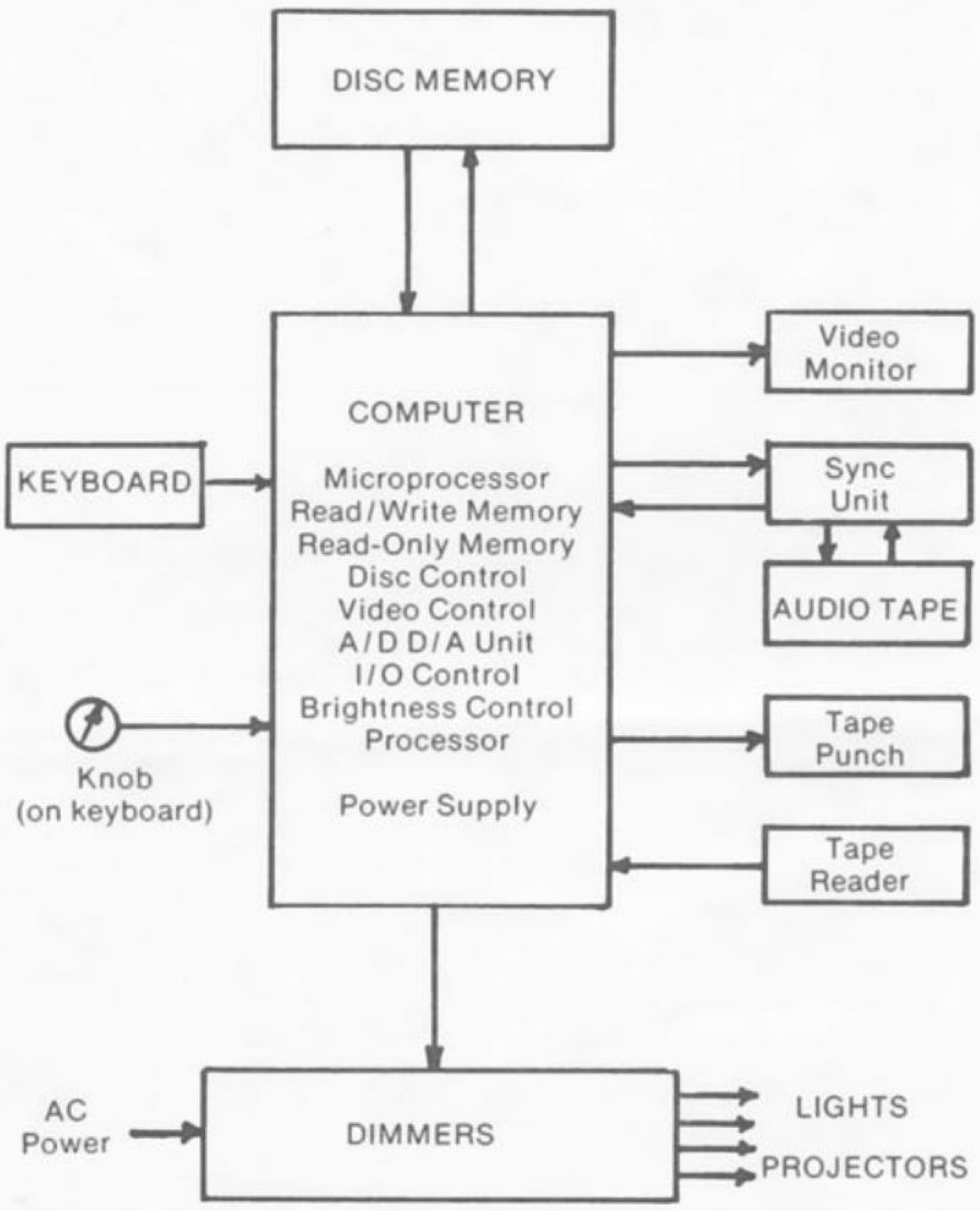


DIAGRAM 2
INTERNAL LAYOUT OF COMPUTER CONSOLE

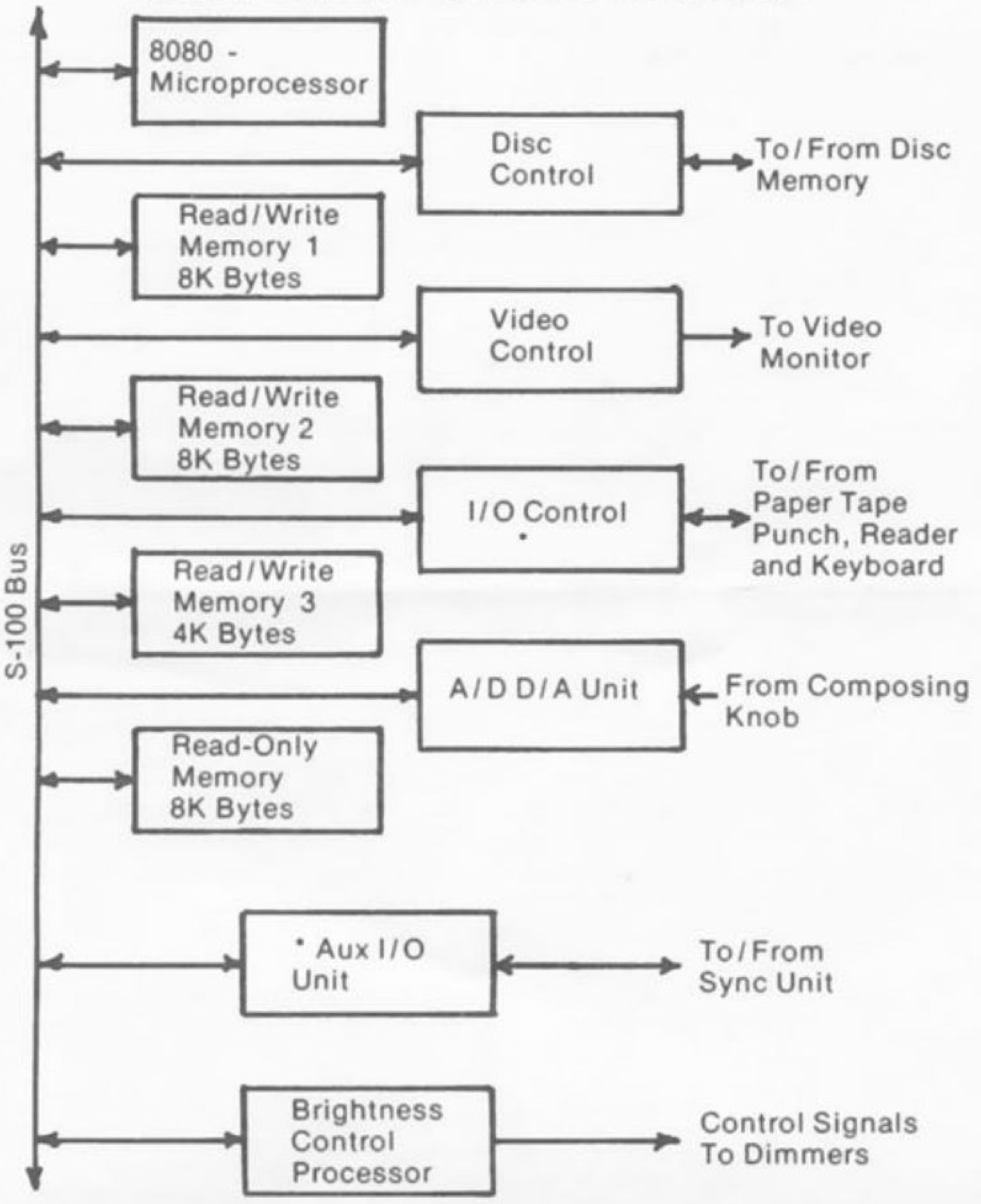
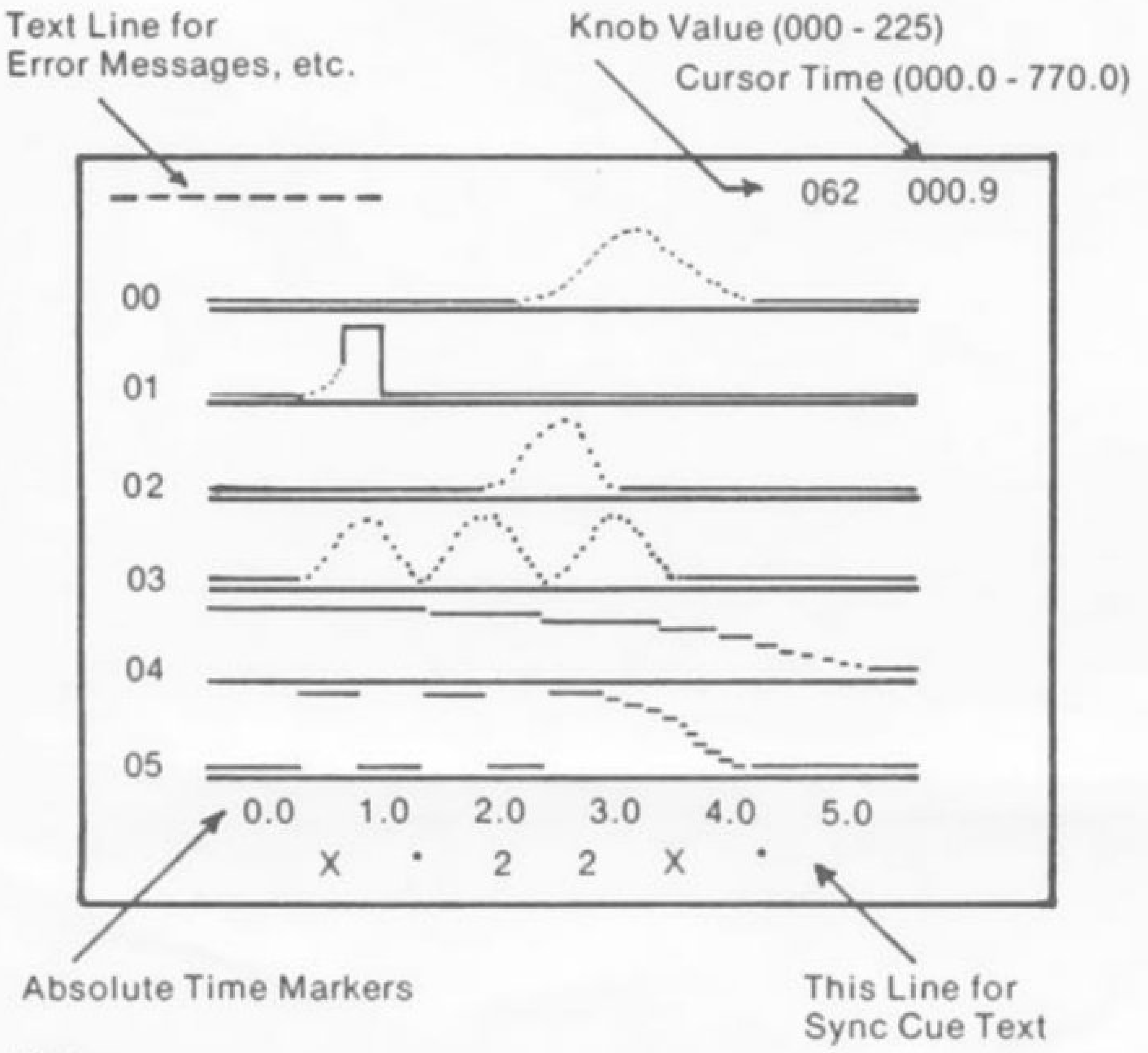


Diagram 3 shows the first "page" of 5 seconds of channels 0-5 and will illustrate some of the composing procedures.

Operating commands are in the form of single-letter commands preceeded by numerical operands. "J" for example jumps the cursor to any time in the 770 second field of the system. Thus "J" alone amounts to "ØJ" and jumps to time zero. "1ØØJ" jumps to 100.0 seconds. "-3J" jumps back 3.0 seconds from where the cursor is now, and so forth.

DIAGRAM 3
DISPLAY FORMAT ON VIDEO SCREEN



Notes:

Channel 0 Shows a fade-up, then down
" 1 Shows cursor at 0.9 seconds entering a fade-up
" 2 Shows a short fade-up, then down
" 3 Shows three repetitions of the channel 2 effect as obtained by the copy feature
" 4 Shows a long fade down, faster at end
" 5 Shows abrupt on/off events, the last fading down

Summary of operating commands:

A assigns channels to the display.
R, L move the cursor right or left in 1/10 second units.
U,D move the cursor up or down a line at a time.
B establishes bounds for automatic (real-time) movement of the cursor and output of current light control data. Bounds may be set for any interval between 0 and 770 seconds.
W asks what the bounds are currently
K turns on knob and allows entering of data either stepwise or in real time.
N causes the numerical value of each function to be displayed in place of the cursor.
G returns to graphic (cursor) display
C copies whatever data lies between the time bounds currently set on the channel where the cursor is located into a copy buffer.
E enters the copy buffer data on to any channel. (The channel the cursor is moved to.) This allows repetitive entering of periodic functions.
F initiates real time running of the cursor between the bounds.
S runs the cursor between the bounds at one sample per second or 1/10 the rate of real time.
O (followed by S or F) causes the real time running of the cursor to repeat indefinitely. "**Return**" cancels or dismisses any command in effect.
X concludes the use of the program and prepares for a return to the program library.

The first revision of this program now in progress will provide two-letter commands in order to add rountines which compress or expand data to fit time limits, rountines which generate periodic functions by mathematical computation, routin which allow sealing or more flexible editing of data.

Finally, it must be realized that at the time the composer is working at the console composing effects, the actual effects are seen in the space where the lights are hung. The graphic display only approximates and summarizes what the composer can see in reality, so the subleties of actual lighting effects are able to be judged and the necessary corrections made with no fear that the schematic transformations of the data which take place at the console will ultimately conceal or confuse the compositional intentions. The greatest benefit of the system is that it allows the most complex and subtle effects to be built up at the rate the composer can manage but retaining exact editorial control over the most minute detail until the desired end is acheived. From that point on the effect is fixed and repeatable, predictable and dependable, yet easily modified or changed to suit a new concept.

Music Catalog

Computer Music Compositions of the United States 1976, a catalog of computer music representing over 100 composers is now available at $5 from Theodore Front, 155 North San Vicente Blvd, Beverly Hills, Va. 90211

AIMS AND MEMBERSHIP

The Society aims to encourage the creative use of computers in the arts and allow the exchange of information in this area. Membership is open to all at £2 or $6 per year, students half price. Members receive PAGE eight times a year, and reduced prices for the Society's public meetings and events. The Society has the status of a specialist group of the British Computer Society, but membership of the two societies is independent.

Libraries and institutions can subscribe to PAGE for £2 or $6 per year. No other membership rights are conferred and there is no form of membership for organizations or groups. Membership and subscriptions run from January to December. On these matters and for other infomration write to Alan Sutcliffe or Kurt Lauckner (U.S.A.)

COMPUTER ARTS SOCIETY ADDRESSES

Chairman: Alan Sutcliffe, 4 Binfield Road Workingham, Berkshire, Eng.

Secretary: John Lansdown 50/61 Russell Square, London WC1B4JX.

Dutch Branch (CASH): Leo Geurts and Lambert Meertens, Mathematisch Centrum, Tweede Boerhaavestraat 49, Amsterdam, Holland.

U.S. Branch (CASUS) Coordinator: Kurt Lauckner, Mathematics Dept., Eastern Michigan University, Ypsilanti, Michigan, 48197 U.S.A.

This issue of PAGE was edited by Kurt Lauckner.

Computer Synthesis of Natural Sounds

Stephen S. Wilson

Introduction

A "natural sound" is that which is created by vibrating objects which occur in nature, such as strings, air columns and three-dimensional objects (bells, etc.). We are used to hearing natural sounds and seldom do we find them unpleasant. Musical instruments (non-electronic) are all natural sounds and will be our primary concern here.

Examples of "unnatural sounds" are those created electronically by "subtractive synthesis" (standard wave shapes filtered and amplitude modulated). A primary reason these sounds do not occur in nature is that all the frequencies are "harmonic", i.e. the overtones of these synthesized sounds are generally integral multiples of a fundamental frequency. In nature almost no object vibrates harmonically. Strings and air columns are approximately harmonic, but it is the slight anharmonics which give natural instruments a reality and warmth as opposed to the "mechanical" sound of an electric piano, for example.

It is not our purpose to propose that sounds which can occur in nature are the only sounds worthwhile synthesizing, but, as we will see, the technique developed to produce natural sounds can be used to create most other sounds, electronic or otherwise, with a higher degree of control.

The Voice Matrix

We will assume that natural sounds can be created with a small number of sine waves (less than 20) with the frequencies independently controlled. Each sine wave will also have an independentaly controlled amplitude of arbitrary complexity. This process is called "additive synthesis". The results are continually changing amounts of energy in each wave as time progresses, i.e. a "spectrum evolution". An excellent review of this technique is given by James A. Moorer (1). Experiments have shown that additive synthesis can produce a piano (2) and trumpet (3) in which even musicians cannot distinguish recordings of the real from the synthetic instrument.

A particular sound or "voice" can be represented by an array of numbers specifying the frequencies and amplitudes of all of the overtones or "partials". The following table illustrates part of such an array of numbers. Time is broken up in 1-millisecond intervals with the instantaneous frequencies and amplitudes given for each partial. Note how easy it is to specify non-harmonic frequency multiples.

TIME →

PARITIALS ↑	1 msec		2 ms	3 ms	4 ms	... etc.
1st freq.	102	Hz	100	100	100	...
1st ampl.	10	Db	30	40	40	...
2nd freq.	208	Hz	201	201	201	...
2nd ampl.	15	Db	35	20	15	...
3rd freq.	310	Hz	305	304	304	...
3rd ampl.	15	Db	40	20	10	...
4th freq.	420	Hz	410	409	409	...
4th ampl.	12	Db	30	15	5	...
etc.	...		...	...	...	...

THE VOICE MATRIX, V. THE NUMBERS IN THIS TABLE REPRESENT ALL THAT IS REQUIRED TO SPECIFY A SOUND.

This array will be called a "voice matrix". The number of values in the voice matrix can be reduced to about 250 by judiciously choosing time intervals other than the 1 ms in the above example and smoothly interpolating intermediate values.

Pitch. A voice matrix will encode the spectrum evolution of one instrument at only one pitch. The timbre at another pitch can differ in several ways. For example, the time scale may change - the decay of the lowest note in a plucked string is about ten times slower than that for the highest note. The relative sound energy in each partial is also different for different notes. In short, all aspects of the spectrum evolution are different and thus each note in a scale would have to represented by a unique voice matrix.

Loudness. If an air column is blown harder or an object is struck harder, the instrument will give, say, double the sound energy. However, that does not mean that all the amplitudes are doubled in the voice matrix because the higher frequency partials are generally excited relatively more than the lower frequencies. Furthermore, there will be more anharmonic distortion due to different average tensions and nonlinearities in an object vibrating with higher amplitudes. Thus separate voice matrices are needed for various degrees of loudness.

Mode. Different modes of vibration can be induced in an instrument by different "initial conditions". For example, a guitar will have a different timbre if the string is plucked close to the bridge or with a pick verses a finger.

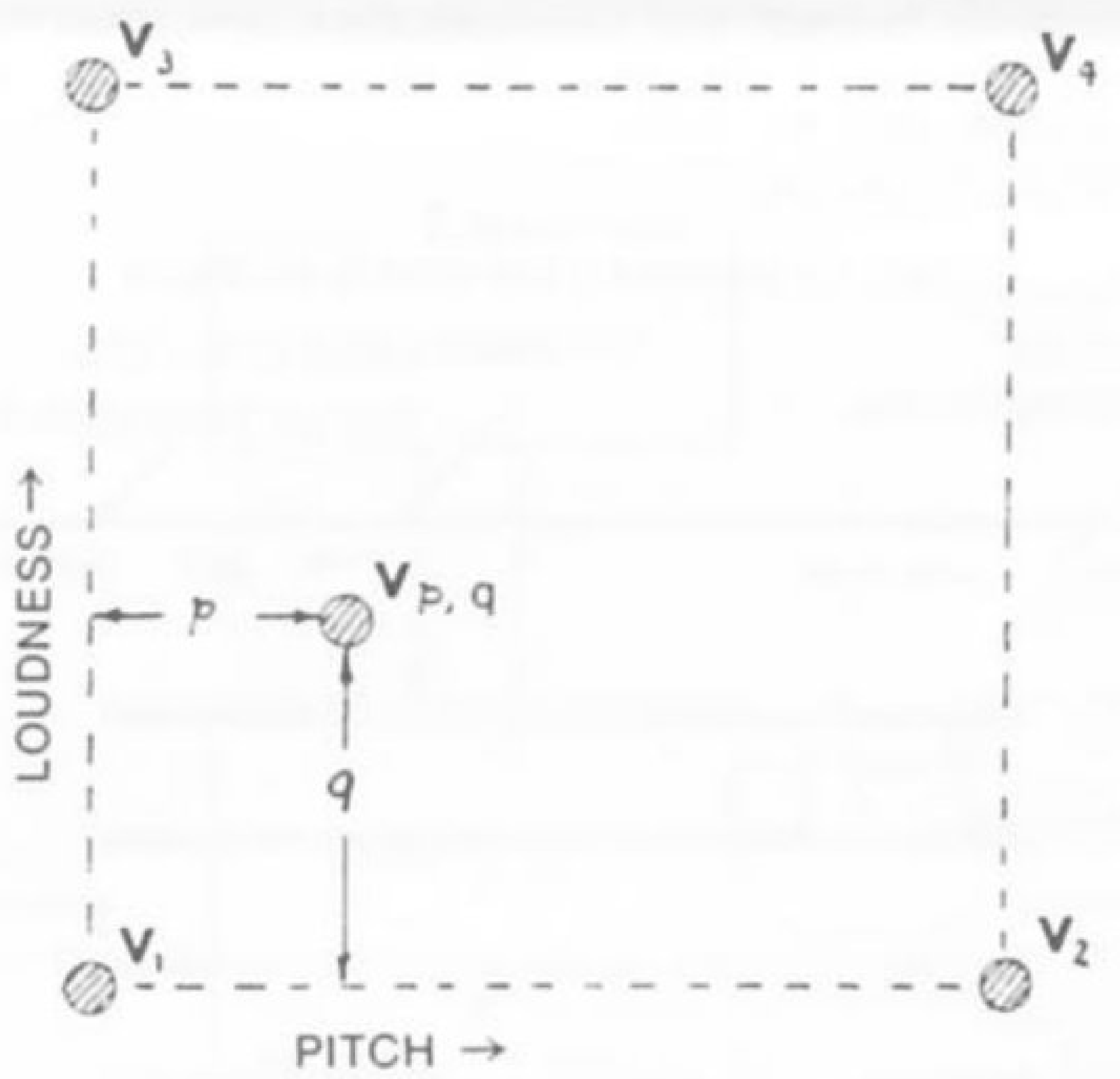
Voice Space

When we consider the voice matrices needed to create all possible combinations of pitch, loudness, and vibration mode, we require an overwhelming amount of data. The scheme that follows will show a method of parameterization which will enable the date to be reduced both numerically and conceptually.

Consider a voice space consisting of a square. The distance along the horizontal will indicate pitch in a scale from 0 to 1; the distance along the vertical indicates loudness on a scale of 0 to 1. Suppose we have only the four voice matrices located at the corners of the square represented by

V_1 = lowest pitch at minimum loudness
V_2 = highest pitch at minimum loudness
V_3 = lowest pitch at maximum loudness
V_4 = highest pitch at maximum loudness.

An instrument playing a particular pitch, p, at loudness, q is located somewhere inside the square.



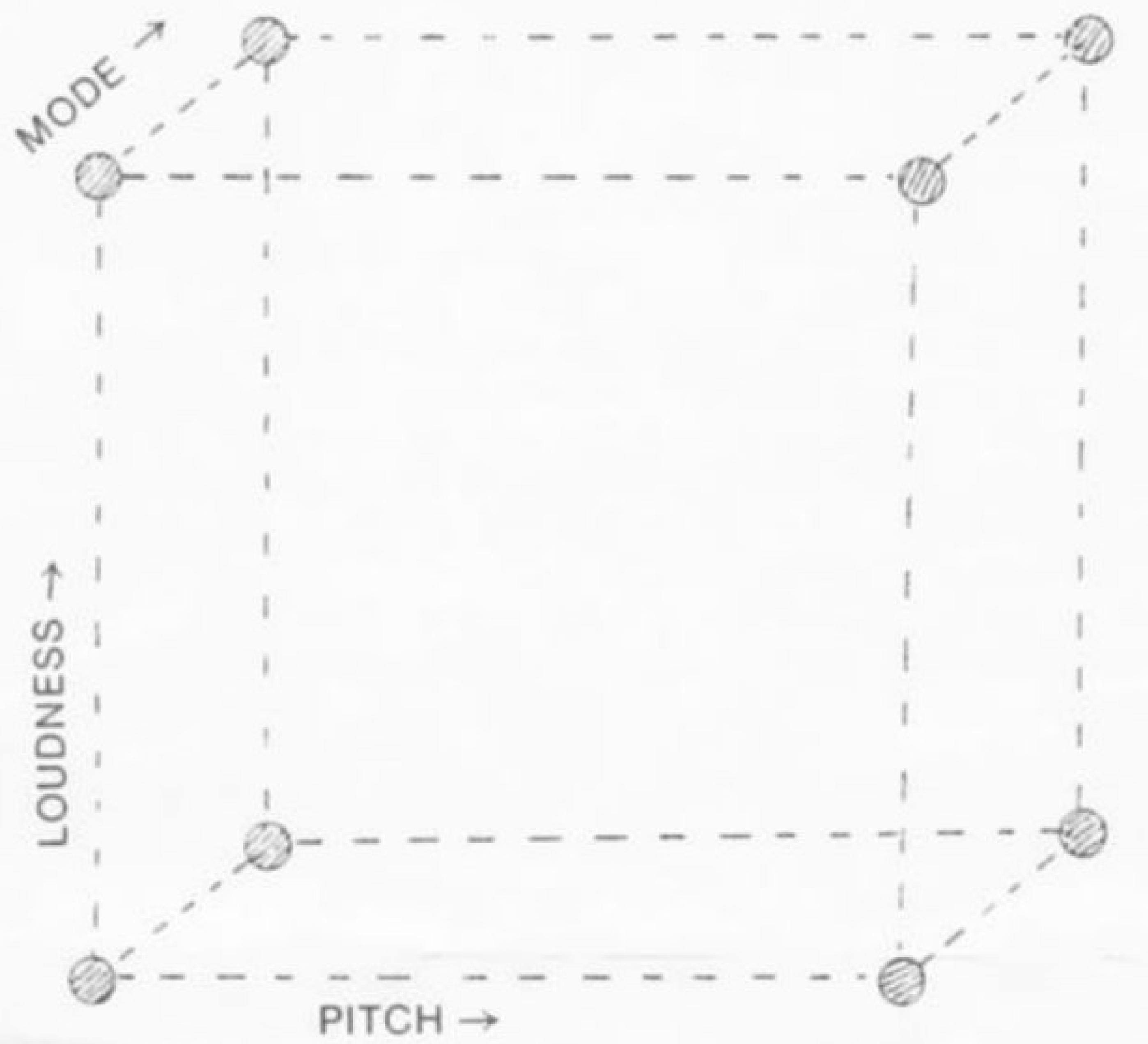
VOICE SPACE, EACH POINT IN VOICE SPACE IS A VOICE MATRIX CONTAINING HUNDREDS OF NUMBERS

We will let the new voice matrix, Vp,q, represented by this note be the weighted average of the four corner voices:

$$\begin{aligned} Vpq = {} & (1-p)(1-q)V_1 \\ & + p(1-q)V_2 \\ & + (1-p)qV_3 \\ & + \quad p\,q\,V_4 \end{aligned}$$

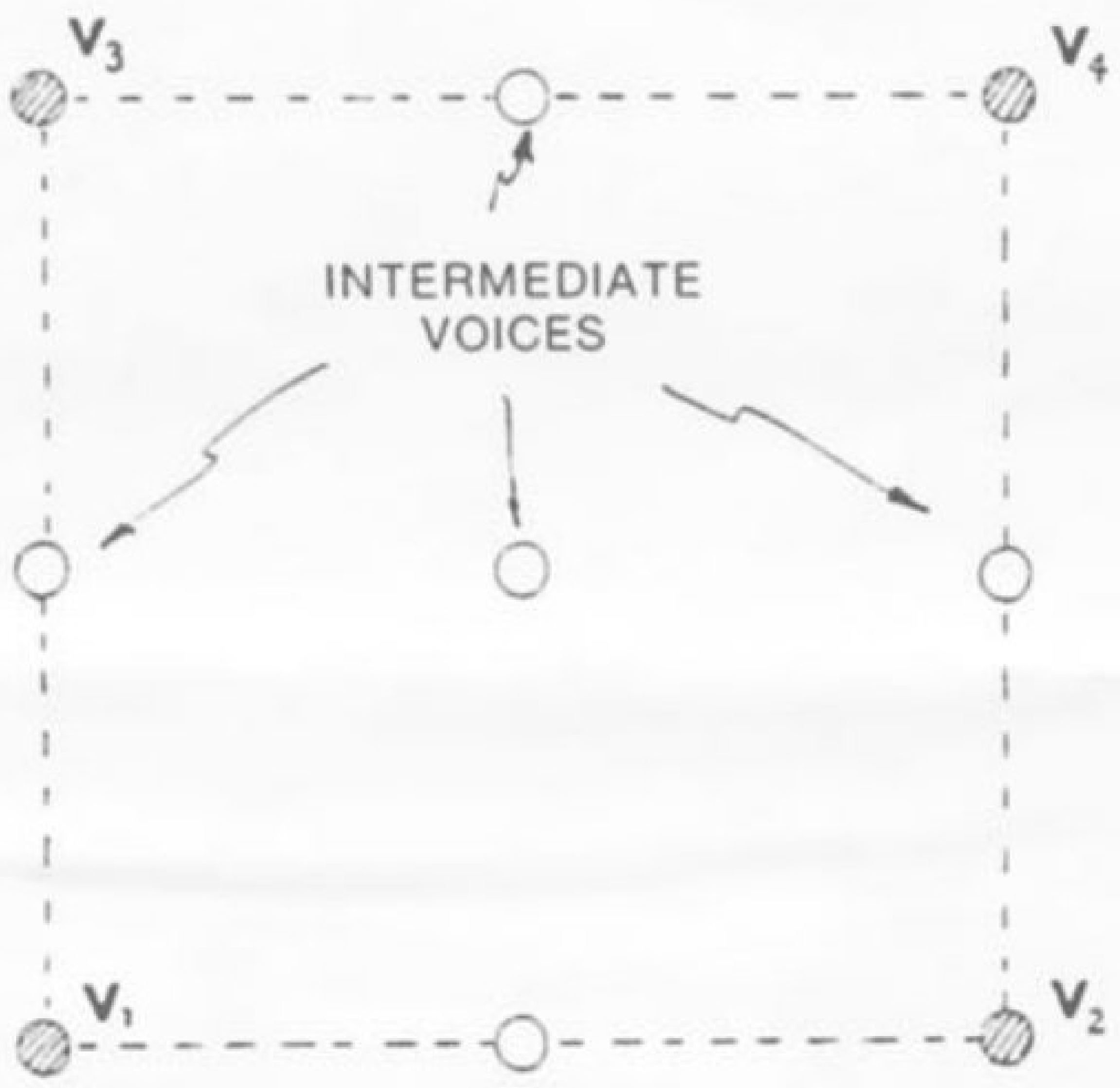
Note that p and q are numbers and the V's are matrices. Each element in Vpq is the weighted average of the corresponding elements in the four corner voices.

To further parameterize the vibration mode, four more voices would be added to form a cube. The weighted average would then contain eight matrix sums.



THE VOICE SPACE CAN BE THREE OR MORE DIMENSIONS. EACH DIMENSION IS A PARAMETER OF THE INSTRUMENT.

One problem of the voice space concept is that the weighted average of the voices is only an approximation of the real instrument. If the approximation is not good enough, more voices would have to be considered to fill the gaps, as shown below.



EXTRA "REAL" NODES MAY HAVE TO BE ADDED TO THE VOICE SPACE FOR A MORE EXACT REPRODUCTION

This technique would have to be used for parameters with wide ranges or for extreme non-linearities such as soundboard resonances.

Summary

In order to synthesize natural instruments a method is necessary to handle the impossible large variety of voices resulting from different pitches, loudnesses, and vibration modes of a physical object. The creation of a voice space in which the dimensions are attributes with which we are familiar helps us conceptualize the synthetic instrument sound and relate it to the real instrument. A simple method of weighted averages of a few critical voices fills in the voice space and optimizes the computer memory allocation.

An additive synthesizer prototype system of 64 sine waves has been completed by the author in a Data General NOVA computer and work is continuing in an effort to increase the number of frequencies and incorporate the above concepts into a microprocessor.

References

(1) James A. Moorer, "Signal Processing Aspects of Computer Music—A Survey," Computer Music Journal, February 1977, pp 4-37. (Preprint of July 1977 article in Proceedings of the IEEE.)

(2) Donnell Blackham, "The Physics of the Piano," Scientific American, December 1975, pp 88-99.

(3) Jean-Claude Risset, Max V. Mathews, "Analysis of Musical Instrument Tones," Physics Today, Vol. 22, No. 2, February 1969, pp 23-30.

The Bangerts

Some Steps Along the Path

Over the years we've had discussions, sometimes severe, sometimes even angry, about why we didn't produce more computer art. A new book would appear, a new copy of PAGE would show up in the mail, and we would say to each other, "let's do more computer art". Then, a month or two later, we would look at ourselves somewhat sheepishly and wonder where our new computer art was.

Most computer art seems to be done on a part-time basis. Knowlton is a research scientist at Bell Labs. Mallary is a professor of sculpture at the University of Massachusetts. We are a programmer at the University of Kansas and a professional artist. Hardly anyone we know who is involved with computer art is able to devote full time to it. This is one of the major reasons why computer art (at least more of ours) doesn't get done.

Part time creation forces us to think very hard about just what we want to do next. Thinking about programs and art, proposing new thoughts and rejecting them, takes up much of our time. This pattern of creation-rejection is not unique to computer art; but the sense of working only part time and the kind of commitment required to write a new program are not always compatible. We reject a lot of ideas. If we were spending 16 person-hours per day at it, then we suspect we would play and experiment more, and spend less time in severe discussion.

Except, of course, everything we're doing is experiment.

Someone needs to write a compiler (interpreter) for graphic language which would be of real interest to artists. We are as guilty as anyone else of continuing to write individual programs each of which requires the kind of commitment and the kind of time of which we have so little.

On the other side are artists who are active in computer art, which is intended to be Art, with a capital A. The quality of art should be judged by artists, critics, informed buyers, gallery owners, art historians, and museum directors. These are the people who have been trained to make valid discriminations between good art and bad art.

Artists don't expect artists to make valid judgements about the quality of programs. Imagine an artist stepping into a programmer's office. "Hi, here's a new computer program I wrote, that I think is really good!", says the artist.

```
DO IT FOREVER;
FROM THE CORNER OF MY WORLD;
FROM SUNDOWN TO SUNUP;
WHILE I CANNOT SLEEP;
STOP;
END;
```

"Uh, artist, that's not a program." "Humph, what do you mean? It's got a DO and a WHILE, and STOP and END, just like other programs I've seen. And who are you to tell me what a program is, anyway. I like it."

Structure Study # III: Black, Brown, Red and Yellow

The programmer may feel that (s)he can decide what is good computer art, but the artist may feel that the programmer is no more competent to judge the quality of computer art than the artist is to judge the quality of computer programs.

Would you like a resolution to this controversy? Wait awhile. Controversies have a way of being resolved or forgotten in a few years. We do not have a pat answer to the question. One of us (Jeff) told an audience of artists in New Orleans that they should stop being hostile to the feelings of ordinary people. Yet, the drawings we do are primarily intended for the educated eye of the person in the art world. Together we have as a goal the production and exhibition of computer art, just as the other of us (Colette) does with her hand-made art.

Our current effort in computer art involves ideas of structure and line. Two examples of drawings are reproduced here. The line is drawn a step at a time, modifying its direction to adjust for smoothness, goal seeking, and adding in a little tenth lagged direction to make it interesting. As always, we are trying to adjust the parameters of the program to make it draw a line which has human, tactile qualities. We expect to present more details in a later paper.

The other idea we're working with is graphic 'structures'. This is a vague idea, connected with the way pictures are put together. We started several years ago thinking about how the computer could be used to solve problems of composition - for instance, where do you place a single figure in a rectangle so that it will have the strongest effect? We did not seem to make any progress.

Last year, we switched to thinking about the placement of figures with respect to each other. This led to the idea of 'pasting' figures together. This is a little like playing with building blocks - small figures can be put together to make larger ones - which can then be put together to make a whole drawing.

We think that 'graphic complexity' is one of the major qualities which can be explored in computer art. Figures as parts of larger figures, interaction of figures in the picture space, repetition and variation - these are ideas which can be manipulated by constructing the appropriate data structures within a program. One can speak of creating such a high level of visual complexity, that the human eye-brain cannot react.

Using our current program, the artist builds skeletons made up of points, by rotating, scaling, and pasting other point-skeletons. The final result is placed in the picture space and the line drawing routine draws from one point in the skeleton to the next. The same point skeleton may be redrawn, using different line parameters and different ink colors.

What we get is a drawing made up of figures which seem to be related, but are not identical. On a larger scale, we can cover a large drawing surface with figures that almost, but not quite, form a pattern.

The examples were all done with the third version of the program. As this is being written, one of us (Jeff) is struggling to get away from the typewriter to continue development of the fourth version, which will include English language parameter input (not quite AGLC) and a simple method for specifying highly complex, irregular figures. It is Saturday, the computer art part of our time.

Jeff Bangert
Colette Bangert
Lawrence, Kansas
October 22, 1977

Structure Study: Black