AN ENVIRONMENT FOR COMPUTER-BASED ANIMATION

B. FERRIS, U. B. C.
D. SEELEY, S. F. U.

ABSTRACT

This paper describes an environment for the production of computer-based animation. The emphasis in development of this environment has been in the production of a habitable environment for a visually sophisticated but perhaps, computer naive animator. The attempt has been to create for the animator/user an environment which allows for flexible and natural ways for not only interaction with the system but for thinking about the processes involved. In response to this, a version of LOGO (called POGO) was chosen as a base language. POGO has SIMULA-like primitives for pseudo-parallel processing as well as a set of graphical primitives. The graphical primitives includes primitives for the tactile specification, manipulation and editing of images as well as primitives for the specification of time-based motion and transformation processes. Using the pseudo-parallel processing primitives, a control structure to allow the specification of parallel time-strands in animation has been developed. The system is a powerful tool both for the use of animators as well as for the investigation of dynamic graphical processing techniques.

UN ENVIRONNEMENT POUR UNE ANIMATION INFORMATISÉE

ABRÉGÉ

Le présent mémoire décrit un environnement pour la production d'une animation informatisée. La mise au point d'un tel environnement visait principalement la production d'un environnement habitable pour l'animateur connaissant toutes les ressources visuelles mais pouvant être novice en informatique. On a tenté de créer pour l'animateur et l'utilisateur un environnement permettant d'employer des méthodes souples et naturelles non seulement pour communiquer avec le système, mais également pour étudier les processus qui interviennent. En réponse à cette tentative, une version de langage LOGO, dite POGO, a été choisie comme langage de base et possède des primitives semblables au langage SIMULA pour le traitement pseudo-parallèle, ainsi qu'un ensemble de primitives graphiques. Celles-ci comprennent des primitives pour la spécification, la manipulation et l'arrangement tactiles des images, ainsi que des primitives pour la spécification des processus de mouvement et de transformation chronodépendants. En utilisant les primitives de traitement pseudo-parallèles, on a élaboré une structure de contrôle permettant la spécification d'une plage temporelle parallèle dans l'animation. Le système est un outil très utile tant au niveau de l'animation qu'au niveau de l'étude des techniques dynamiques de traitement graphique.

AN ENVIRONMENT FOR COMPUTER-BASED ANIMATION

Introduction

Computer Graphics facilities for the production of animated sequences have been in existence for some time. Numerous researchers have designed and implemented computer systems of varying complexities to facilitate the production of animated films. Many of these systems, though they provide the appropriate production capabilities, still suffer from the problems inherent in much of the field of computer graphics.

That is, it is difficult for someone not versed in the systems programming for that graphics system, to have full and independent control over the creation of animated sequences (strands). One exception is GENESYS (2) and related systems being developed at Xerox Palo Alto Research. Rather than concentrate efforts on the development of a production facility, an attempt was made in the UBC system to produce an environment which would avoid advanced systems knowledge and any systems awkwardness, yet be a flexible enough medium for development and exploration of new graphical techniques and image sequences by the artist or educator as a non-programmer. Production of film and/or video-tapes, however, was still of major concern. The hardware used consists of an ADAGE-10 graphics computer with light pen, data tablet, function buttons, joystick, etc., hooked by means of a fast line (100K Band) to a 370/168 running under MTS. All filming is done directly off the CRT face.

The Approach

There is a disparity in many computer graphic systems between the manner in which the user must interact with the system, and the manner in which the user would like to interact with the system. This problem is heightened in the area of animation in that most users, although perhaps naive in regard to computer systems, are visually quite sophisticated. This came immediately to light when we first began to interact with animator/artists. The initial contact with already existing hardware and software on the UBC ADAGE graphics system was a frustrating and traumatic experience. Firstly we ran into a vocabulary problem in merely communicating disparate views of the same visual phenomena. While our dialogue was full of reasonably technical jargon, the animators talked in a much freer manner. Their speech was full of adjectives and tactile references.

Secondly, although excited and stimulated by the visual presentation, there was an element of frustration associated with

the initial contact. The animators were asking "Can the computer do . . . ?" Usually there were two typical responses; either "Well, yes but I'd have to reprogram . . ." or "I can do something like that . . ." for example a common desire was to add more "texture" to the line images. From this sort of interaction, we realized that there were a number of major things that an animation system must do. (For other experience with artists, see (6) and (7).)

First any animation system must be a habitable environment.(5) Habitability being the quality of a system that makes it confortable to learn and use. Secondly, the animator must have complete control over the system. Many current animation systems require extensive programmer-animator collaboration.

Often then, the product becomes the programmers idea of what the animator thinks the computer can do. To avoid this sort of man-man-computer relationship, a programming environment directed at the animator would seem necessary.

There are other animation systems that place the emphasis on implicit, programmed-in strategies. To allow more animator control, an animation system should not force strategies, but supply tools for the development of strategies.

An animation system should allow for real-time manipulations and interactions. The feeling of immediacy in interaction is very important; by allowing reviewing, manipulation, and editing ability over images and sequences, the animator's opportunity to improvise and explore their intuition is greatly enhanced. Since in most animation, it is sufficient for things to 'look right' (as opposed to being right), the use of tactile adjustments of images and processes is very important.

The system should be tactile, in that images and motion sequences are synthesized and manipulated by an appropriate choice of dials, joysticks, tablets, light pens, etc. As Bob Futrelle has said "You don't type in 'FORK' before picking up a piece of food/" The best way for a tool to be used as an extension of the person is to place it in their hands, and not have the creative control have to pass through their symbol-manipulating intellect. With computer animation, as evidenced by "La Faim ", this will result in more evocative and "human" images, rather than the over-geometric style of much computer animation (4). In order to open the gates to creativity and intuition, the tactile is preferred to the verbal, and the analogue is preferred to the digital!

Because of the exceptional range of languages and software development features in the MTS environment, we wished to embed an animation facility as a special environment in the interactive and extensible language, LOGO. Using ideas from simulation (3) (8), and the language SMALLTALK developed by Alan Kay, a "parallel LOGO" called POGO, incorporating pseudo-parallel processing and a special inter-process communications facility, was designed for this purpose. Concurrent to its ongoing development, a menu-oriented, heavily

tactile, command language version called PEACH was implemented in order to gain more experience in providing access by artists to computer animation tools. We hope in this manner to further refine the design of POGO.

LOGO was chosen as our target language because it was originally designed for young and informal programmers. Our approach does not bear any resemblance to that of Newman's . It exhibits many of the properties of a habitable system:

- *LOGO is designed for 'symbolic processing' rather than 'numeric processing'.*
- *LOGO is an extensible language thus allowing the user to build up his own vocabulary of processes for dealing with the system.*
- *LOGO emphasises functional and structural programming in its central structure.*
- *LOGO is (comparatively) easy to learn to use.*
- *Experience with LOGO systems has shown LOGO to be a valuable tool for novice computer users. That is, significant results are possible in a relatively short period of time.*
- *LOGO allows the easy addition of various 'micro-worlds'.*
- *LOGO is a 'friendly' environment.*

Creating Animation with PEACH

*In keeping with our "tactile strategy" it was decided to have PEACH run as entirely on the ADAGE display screen as possible, avoiding the disruptive process of forcing the user's attention to switch back and forth to the 3270 MTS terminal in order to process commands. This is accomplished by having the system driven by menus and by echoing the 3270 keyboard. Many of the animation terms in what follows are taken from Baecker (**1**).*

The screen is effectively divided into three areas (see Figure 1). The message area is used to prompt the user, to remind the user of system status, and for echoing the input from the keyboard. The menu area contains a window which displays a selection from the current menu. The user 'rolls' over the menu alphabetically using a sliding potentiometer, and upon finding the desired item indicates this with a button. The menu was implemented in this manner first to save on screen area, and secondly to save on the number of vectors being displayed (the ADAGE is inefficient with character vectors). Since there is a small fixed number of commands, and a relatively simple tree structure to the menus, these need not be continuously

displayed. The user can "thumb" through the list easily. Finally, all user generated graphics are shown in the relatively large display area. Previewing animation sequences is done with the entire screen.

There are three levels to the PEACH command system. The first is the Cel level where the creation, manipulation, editing of individual images, and image composition occurs. Next is the Strand level where the creation, editing, and previewing of the individual motion (dynamic) sequences occur. The last is the Movie level where a "movie" can be composed, edited, and previewed from various overlapping strands. The user may freely move between levels using current image processes or restoring those that have been saved in a library. The details of these command levels follow:

The Cel Level

Individual static images are called "cels" and may be created by 1) sketching on the data tablet, 2) fetching library images, 3) manipulation and/or composition of existing cels. The commands in the menu list below act on cels . . . cels are named at their time of creation. The display screen for a typical cel level interaction is shown in Figure 1.

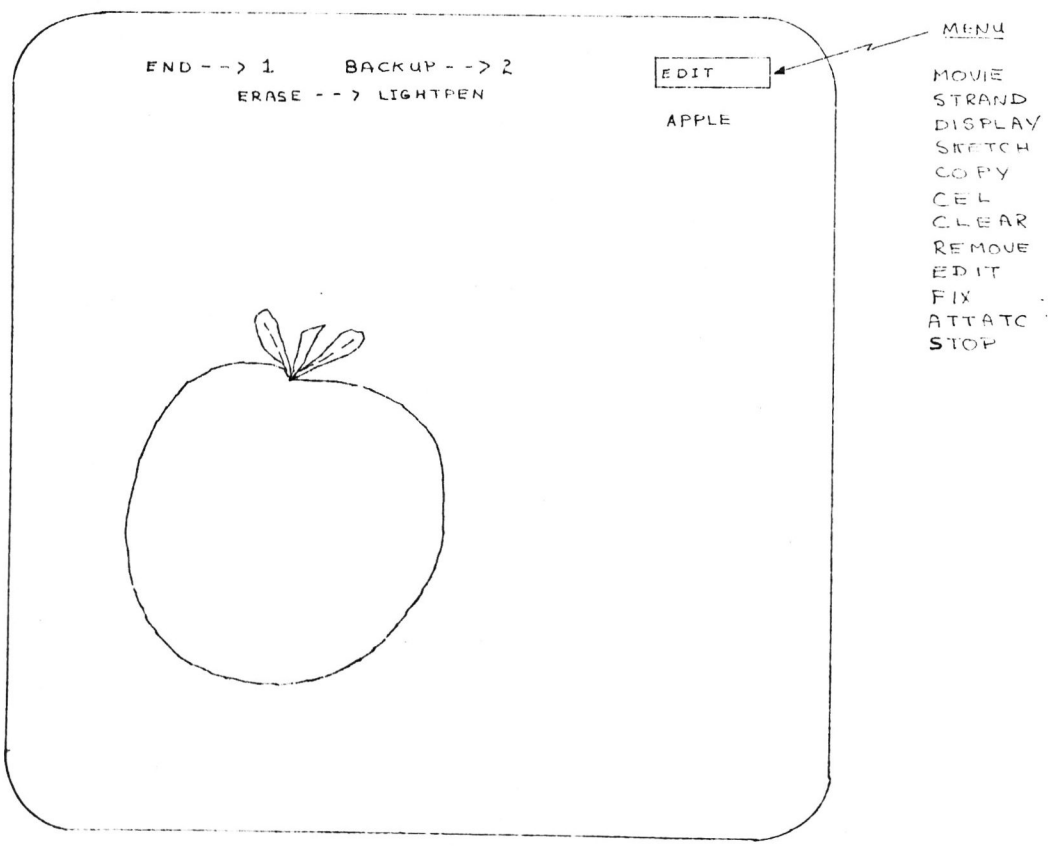

FIG 1 DISPLAY & MENU FOR CEL LEVEL

Menu for Cel Level

ATTACH
CEL
CLEAR
COPY
DISPLAY
EDIT
FIX
MOVIE
REMOVE
SKETCH
STOP
STRAND

Most of these require a Cel name as an argument, this may be provided by rolling through the Cel library using the sliding pot or by keyboard specification, suitably prompted and echoed.

Both STRAND and MOVIE transfer menus to their respective levels while prompting for the name of a new/old strand or movie. STOP ends the session and prompts for which created cels are to be kept until next session. CLEAR wipes the display area.

CEL prompts for a new cel name. It takes the compositon consisting of all currently displayed cels in the displayed states and position and creates a new cel for the library. It fixes cel copies before composition and is the only way to compose images; hence PEACH assumes that visual cues are necessary for composition.

SKETCH also prompts for a new cel name and displays a message informing the user to go to the tablet. Sketching takes place in a natural fashion and may be terminated by touching the word END as reflected on the screen. The last drawn vector may be erased by similarly touching the word BACKUP.

COPY prompts for first the original cel name (omitted if copying image already on screen) and then for the cel name of the copy (which may be omitted). It produces and displays a copy of the original cel. This copy may be transformed by inputs from various dials for rotation, brightness, scale, x-translation, and y-translation, allowing tactile manipulation of an image. The copy remains attached to the dials until another command implicitly or explicitly "fixed" it. FIX detaches the current cel copy from dial manipulation; a new cel name may be given at this time. ATTACH provides the same function as COPY except that a new cel is not created (i.e. the original is being manipulated).

EDIT displays the appropriate cel and allows selective erasure of vectors by light pen hits, alteration by rubber-banding, and vector BACKUP. A copy of the original cel is attached to the dials. When END is hit the copy is FIXed and replaces the original cel.

In composing the semantics of these commands as much tactile specification as possible has been encouraged. A further reduction in keyboard work is attained by placing the cel dictionary, alphabetically under sliding pot selection.

The Strand Level

On this level motion sequences can be created and stored for future composition. Each time STRAND is selected the user is prompted for the strand name. If it is new, then the user is prompted for the name of an initial cel of a strand. Any of the motions commands may be chosen (or rechosen) in any order in order to specify the motion parameters of this specific strand. A typical screen for this level is displayed in Figure 2, and the menu selections are as below:

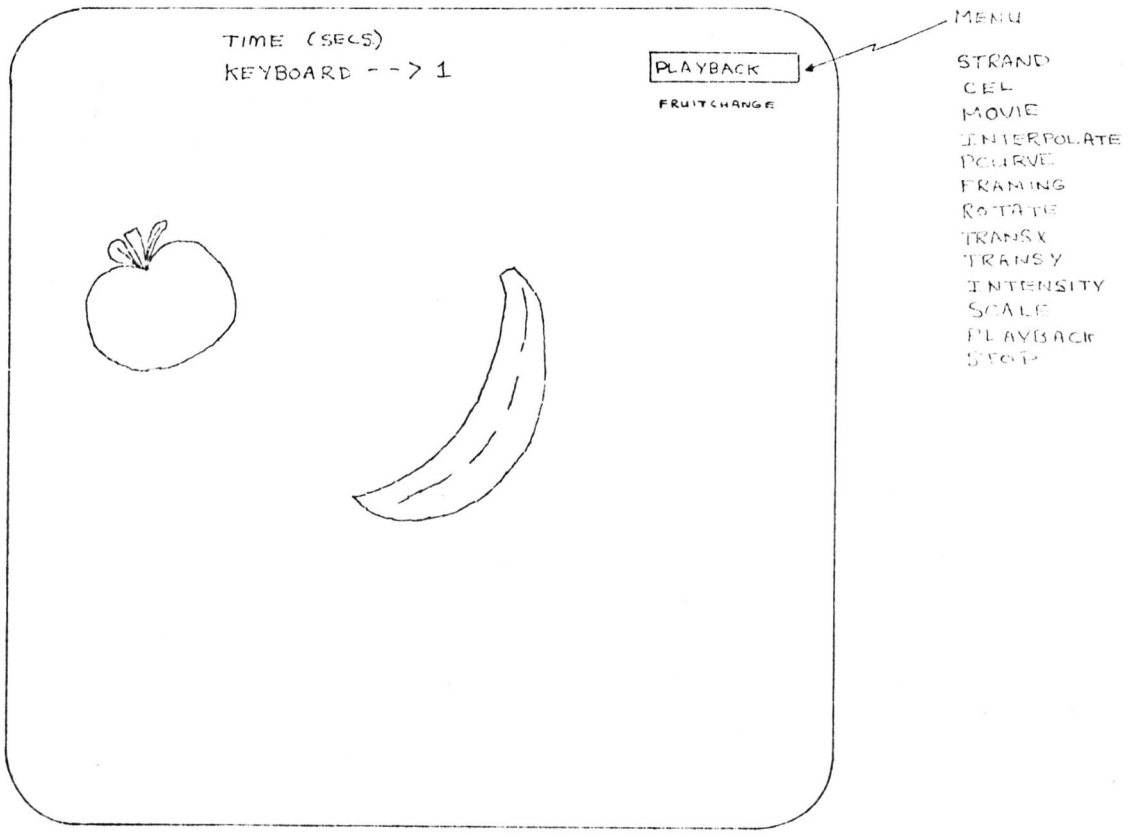

FIG. 2 DISPLAY & MENU FOR STRAND LEVEL

Menu for Strand Level

CEL
CONTINUE
INTENSITY
INTERPOLATE (PCURVE,FRAMING)
MOVIE
PLAYBACK
ROTATE
SCALE
STOP
STRAND
TRANSLATE X
TRANSLATE Y

The order of transformations is:
INTENSITY (TRANSY (TRANSX (SCALE (ROTATE (INTERPOLATE (X))))))

INTERPOLATE prompts for the name of the cel into which the initial cel will be transformed; as a side effect it adds PCURVE and FRAMING to the menu. If PCURVE is never selected then the interpolated vectors will have their end-points more in straight lines during interpolation. If it is selected however, the user is prompted for cel name of the median path along which the vectors will move. If FRAMING is not selected, the interpolation will proceed linearily. That is, the interpolation paths are sliced into equal increments. However, if it is the user then must select between LINEAR, ACCELERATION, and DECELERATION in the message area, when the latter provide the interpolation equivalents to fade-outs and fade-ins.

PLAYBACK causes the current strand to be previewed; the screen clears and the animation sequence takes place. It prompts the user for a time value in seconds (each frame is 1/40 second) of the strands' duration. After playback, the user is still at the strand level and may alter the strand definition and iterate or CONTINUE edit any of the cels and return, or compose a movie. CONTINUE takes the last cel of the current strand and uses it as the initial cel of a continuing motion sequence.

All of the remaining commands at this level produce an additional display image of the initial cel with only the appropriate function attached to a dial. For example, if ROTATE is chosen a 'copy' of the initial cel is produced with only the rotate dial active; a digital readout of this value will appear upon the screen (-1 to +1). A special RATE dial may be altered giving another digital readout (0 -- 20) giving the number of times the specified transformation is to take place during the strand's duration. For example, if the rotation dial value is .5 and the rate dial is 10 then 900^{o} of rotation would take place during the strand (i.e. 90^{o} every 1/10th of the time duration).

Upon fine tuning a strand, the user may wish to include it amongst other strands in a movie composition.

The Movie Level

CEL
COMPOSE
FILM
MOVIE
PLAYBACK
STOP
STRAND

Upon entering the movie level, a display such as that in Figure 3 will appear. Its purpose is to provide the user with a tool to compose separate time strands into a MOVIE allowing the

visual-tactile synthesizing and synchronization of various time strands. COMPOSE prompts for a strand name, making it the currently active strand. Each strand is displayed as a horizontal bar graph, the ends of which can be manipulated by the top two dials. A strand can be reactivated by pointing the light pen at the appropriate name beside the bar graph. Vertical lines are used as alignment guides in order to synchronize motion events. PLAYBACK prompts for the movies duration as in the strand level. FILM displays the current movie frame by frame flicking a relay that drives a movie camera (assumes the latest PLAYBACK timings).

Summary

An interactive system called PEACH has been described for the composition of complex animation sequences that are picture driven. It is very tactile in its approach and does not demand that the animator know any programming yet he/she may carry out their composition entirely under their control.

At this time the authors would like to acquire more experience working with artists in both a tactile system such as PEACH and the procedural-oriented approaches such as in parallel LOGO before deciding how to best marry the two approaches, if at all.

Finally, we feel that systems such as PEACH that open access to computer graphics to artists, educators and other visual communicators will broaden the scope for more humanizing applications of computers in general. Furthermore, the opportunity to use advanced technology in a manner that liberates the creative and intuitive abilities latent in the right brain hemispheres of symbol-sodden technocrats could help provide a happier society.

Bibliography

(1) *Baecker, R.M. "Interactive Computer-Mediated Animation", Ph.D. Thesis, M.I.T., 1969.*

(2) *Baecker, R.M., et al. "GENESYS -- Interactive Computer-Mediated Animation", in Computer Animation, Hastings House, 1974.*

(3) *Baecker, R.M. and Horsley T.R. "Computer-Animated Simulation Models. A Tool for Transportation Research", to appear in Transportation Research Record.*

(4) *Burtnyk, N. and Wein, M. "Towards a Computer Animating Production Tool", Proceedings of Eurocomp Congress, 1974.*

(5) *DeFanti, T.A. et al., "Computer Graphics as a Way of Life", Proceedings of First International Conference on Computer Graphics and Interactive Technique, Boulder Colorado 1974.*

(6) *Knowlton, K. "Collaborations with Artists - A Programmer's Reflections" in Graphic Languages, edited by Nake and Rosenfeld.*

(7) *Mezei, L. and Zwain, A., "Arta, An Interactive Animation System", Proceedings of IFIP 1971.*

(8) *Seeley, D., and Walker, W. et al. "Interacting With Discrete Simulation Using On Line Graphic Animation. Graphics Conference, Boulder Colorado, 1974.*

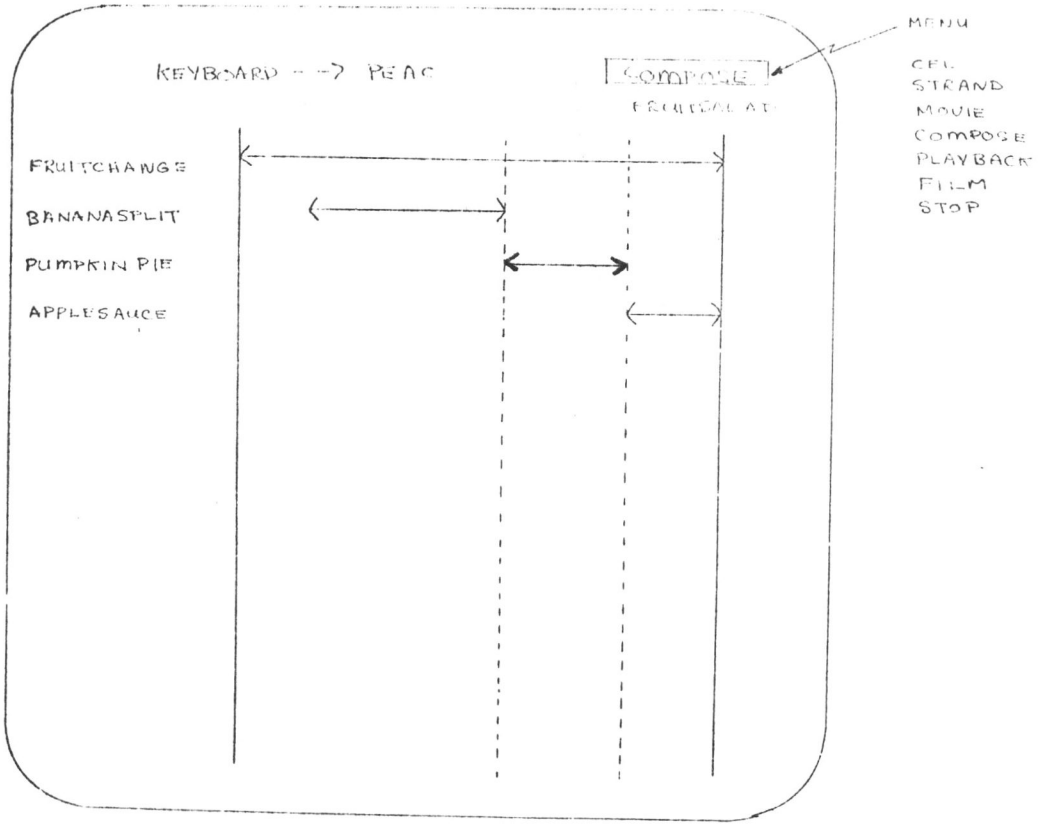

FIG. 3 DISPLAY & MENU FOR MOVIE LEVEL