A COMPUTER ANIMATION FACILITY FOR RESEARCH AND EDUCATION FILMMAKING: DESIGN AND APPLICATION

RONALD M. BAECKER
UNIVERSITY OF TORONTO

ABSTRACT

A comprehensive computer animation facility for research and educational filmmaking is described. The facility is composed of an animation package (ZAPP) which is callable from several high-level languages and which has a rich set of tools for defining and controlling movement and picture change, a real time playback and editing system, a free-hand sketching system, and a movie postprocessing system, which ultimately records the movies on film. The paper discusses the major features of ZAPP, then summarizes 18 months of its use in artistic and educational filmmaking, visual simulation and the depiction of complex processes, and computer program illustration.

SERVICE D'ANIMATION INFORMATISÉE POUR LA RECHERCHE ET LA CINÉMATOGRAPHIE ÉDUCATIVE: CONCEPTS ET APPLICATIONS

ABRÉGÉ

On y décrit l'ensemble du service d'animation informatisée destiné à la recherche et à la cinématographie éducative. Ce service comprend un programme-produit d'animation (ZAPP) qui peut être employé pour plusieurs langages de niveau élevé et qui possède un ensemble élaboré de moyens destinés à définir et à contrôler le mouvement et le changement d'images, un système de reproduction et d'arrangement en temps réel, un système de dessin à main levée et un système de traitement ultérieur des films qui en fin de compte enregistre le film sur pellicule. Le mémoire discute des principales caractéristiques de l'ensemble ZAPP, puis fait le résumé de 18 mois d'utilisation dans les domaines de la cinématographie artistique et éducative, de la simulation visuelle et de la description de processus complexes, et de l'illustration de programmes d'ordinateur.

A COMPUTER ANIMATION FACILITY FOR RESEARCH AND EDUCATIONAL FILMMAKING: DESIGN AND APPLICATION

Ronald M. Baecker

University of Toronto

I. Introduction

Late in 1972, we began the design of a new computer animation system. Our goal was to develop a comprehensive facility that could support a broad range of research and educational filmmaking in the University of Toronto community. As the design evolved, the following constraints, assumptions, and intentions were paramount:

1) Campus computing centered chiefly around a large batch processing machine. The machine could not support a responsive and cost-effective interactive graphics system.

2) There was developing on the campus an increasing trend towards the acquisition of minicomputers, some able to provide inexpensive interactive graphics on a small scale. However, these machines lacked significant numerical processing capabilities, on-line file storage, and high-level language support.

3) The greatest potential market for a campus computer animation resource seemed to be among individuals with programming skills rather than individuals with drawing skills. For instance, we wanted to make it easy for researchers to translate experimental results into visual terms.

We therefore decided to build an algorithmic animation system, in which one could construct movies by writing programs, rather than a demonstrative system, in which one could draw movies. Furthermore, the system had to be accessible from popular high-level languages running on the large batch machine. The resulting design included a major system running on that machine and three auxiliary programs running on the minicomputers:

1) The Zoom-Animate-Pan Package (ZAPP), a FORTRAN-PL/I-ALGOL/W callable system with a rich set of tools for motion specification and control, and many practical aids for the working animator (1);

2) the Film Reviewer and EDitor (FRED), which allows real-time playback of, experimentation with, and editing of ZAPP movies (2);

3) the Freehand Input To Zapp system (FITZ), which allows sketching of images and movements for later transmission to and use by ZAPP (3); and,

4) the Film Enriching Recorder and Developer (FERD), which adds line thickness and region shading, accounts for the effects of transparency and opacity, and produces colour separations while driving a microfilm recorder.

The intent of this design is that the animator develop and debug a rough version of her film using ZAPP, experiment with its dynamics and timing using FRED, input any needed free-hand sketches with FITZ, and produce the polished footage with FERD. All of the programs communicate using a common data format.

ZAPP has been operational since late summer, 1973. A preliminary version of FRED came up in May 1974; the complete design is described in (2). FITZ was written in late summer, 1974. FRED has not yet been implemented; a rudimentary microfilm plotting package has been used in its place.

The remainder of this paper discusses the major features of ZAPP, then summarizes 18 months of its use in artistic and educational filmmaking, visual simulation and the depiction of complex processes, and computer program illustration.

II. Movement and Picture Change Descriptions in ZAPP

ZAPP's major strength as a computer animation language stems from its rich set of tools for describing movement and generating picture change.

A. Basic Movement and Picture Change

In ZAPP, a picture is composed of curves, each of which consists of individual points or sets of points connected by straight lines. ZAPP allows the animator to construct images with procedure calls that create and open a new image, draw points and lines at specific coordinates and add them to that image, and close and store the image. There are also mechanisms to read in an image description from cards, read it in from a file prepared by the sketching system FITZ, retrieve it from an image library, or create it as one of ZAPP's basic set of algorithmically-defined geometric figures, such as circles and polygons.

These images may then be moved, changed, and combined with the folowing mechanisms:

-- a program that interpolates, or generates automatic in-betweens, from one image to another;

-- a program that moves one image along a dynamic picture path, or p-curve (4), also defined as a ZAPP picture;

-- a rich set of functionally controlled geometric transformations;

-- a "generator" program which allows the image to be dynamically revealed or covered; and,

-- any algorithmically describable transformation.

1. Interpolation

ZAPP's interpolation method is the same as that used by other investigators (5). Although cumbersome, it can produce striking results in the hands of a skilled and careful animator (6). Two key pictures, assumed to contain equal numbers of curves, are compared in terms of constituent points, and the curves having fewer points reconstructed with more added. Corresponding line segments in each curve are then interpolated over a number of frames, with the rhythm (See Section II.A.3.) of the interpolation determined by a function.

2. P-curves

Although the p-curve is most useful in an interactive animation system accepting free-hand input(7), it has applicability for ZAPP as well. The p-curve is an image that describes both the trajectory and the dynamics of movement of another picture. For example, using as a p-curve a circle with defining points not evenly spaced creates a circular movement with a speed determined by the spacing of the points. The ZAPP animator can cause any picture to be treated as a p-curve for any other picture.

3. Functionally controlled transformations

All graphics systems contain geometric transformations such as translation, rotation, and scaling. ZAPP contains a rich set of such functions:

-- movement to an absolute position;
-- movement by a certain amount;
-- movement relative to another picture;
-- rotation about a picture's center;
-- rotation about a specific point;
-- rotation about another picture;
-- scaling a picture about its center;
-- scaling a picture about a specific point;
-- scaling a picture about another picture;
-- scaling a picture to fit in a rectangular window;
-- centering a picture in a rectangular window; and,
-- changing the intensity of a picture;

These functions can be applied either "statically" or "dynamically". Under static processing, a picture transformation maps an old picture into a new one, but the change does not appear directly on the film. Static processing is like arranging a cast of characters on the stage before the play begins. Under dynamic processing, the picture transformation executes as part of the movie. Dynamic processing is like the actions of the characters within the play.

Consider, for example, a typical ZAPP movement:

```
CALL MOVETO (PIC,FLINR,13.,FACDE,10.)
```

The final result of this transformation is to move the picture PIC to X=13.0 and Y=10.0. The functional parameters FLINR and FACDE determine the rate of change of PIC's location with respect to time within a prescribed interval. (See Section II.B.2.) In particular, the X coordinate changes linearly from its current value to 13.0, while the Y coordinate changes to 10.0 in a smoothly accelerating then decelerating fashion. Thus each application of a transformation can result in a motion with its own unique rhythm.

Transformations can be driven by two kinds of functions, periodic and aperiodic. Since both new transformations and new functions can be added to the system by a knowledgeable user, a standard interface is defined. Functions are assumed to accept temporal fractions between 0 and 1, and return values between -1 and +1. Periodic functions must return 0 for time=0 and time=1; aperiodic functions must return 0 for time=0 and 1 for time=1. It is the responsibility of each transformation to map frame numbers into temporal fraction between 0 and 1, and to map fractional movements into absolute changes in position, orientation, and size.

4. Generators

An effective attention-drawing animation technique is to bring in a new image by uncovering successive portions of it. A graceful way of removing an image from the screen is to wipe out successive portions of it. The ZAPP generator transformation allows any portion of an image to be drawn or erased at a rhythm determined by a standard ZAPP function. Appealing visual effects can be achieved by thoughtful choices of the segment of the image, the polarity of the generator (in or out), and the rhythm of the change.

5. Algorithmically defined sequences

The above set of techniques is quite comprehensive compared to many computer animation systems, but there are still some animated sequences that cannot gracefully be constructed. The ZAPP animator can always fall back upon the capabilities of the host language and include in his movie any computable sequence of images.

B. Movement and Picture Change Combinations

Additional tools are provided for the ZAPP animator to combine these primitive movements and picture changes into more complex sequences.

1. Absolute and Relative Motions

ZAPP has one absolute coordinate system which spans all of real two-dimensional space. All coordinate references are defined with respect to this coordinate system. Relative motions of a picture can be produced with a set of transformations which are expressed relative to the location of the "center" of another picture. The center is usually considered to be the midpoint of a minimum enclosing rectangle, but a new center may be established by the user at any time. Relative motion descriptions can be concatenated to arbitrary depth providing they contain no loops.

2. Concurrent Motions

A computer animation must facilitate the description of concurrent actions, involving both synchronous and asynchronous events and movements. Several movements must be simultaneously applicable to a single picture; several pictures must be transformable in parallel. Despite its importance, no programming-language based animation system prior to ZAPP has provided this capability.

A call on the ZAPP INTRVL routine establishes a time interval over which all following transformations will apply. Transformation requests are not processed immediately but rather stored in an <u>agenda</u>. The agenda records the key to the type of transformation, the picture to which it will applied, the transformation arguments needed for the calculation, and the beginning and ending frames of the interval. New intervals can be established with additional calls on the INTRVL routine, hence both synchronous and asynchronous motions may be described. When all the motions for a portion (T1, T2) of the film have been specified, a call ROLLIT (T1, T2) causes the processing of all transformations stored in the agenda and applicable between T1 and T2. All pictures referenced by these transformations, and any number of unchanging "background" pictures, are automatically output in the correct frames. Interpolation, p-curve driven movements, functionally controlled movements, generators, and algorithmically described sequences, can all be added to and processed concurrently from the agenda.

3. Cameras, Windows, and Viewports

While all these concurrent actions occur on the "stage" of the single ZAPP display page, four independent "cameras" can record different portions of the scene. Each camera is pointed at a rectangular window on the display page, and can be panned and zoomed over the page to change the location and size of the window. After the scene is clipped to the boundaries of the window, its contents are mapped onto a rectangular viewport, or portion of the movie screen. Thus split-screen presentations can be generated very easily.

III. Facilitating the Movie Production Process in ZAPP

A. Multi-lingual use of ZAPP

90% of the jobs run at the University of Toronto Computer Centre are written in PL/I, FORTRAN, or various of their dialects and subsets such as WATFIV, PL/C, or SP/k. We have therefore made ZAPP directly accessible, via procedure calls, from these two languages. We also sought to make the package callable from more elegant high-level languages, and have achieved this, albeit somewhat awkwardly, with ALGOL/W.

B. Use of multiple devices in ZAPP

The ZAPP animator can easily direct output to a line printer, paper plotter (ink or electrostatic), or microfilm recorder. One saves time and money by debugging the basic algorithms on the line printer, then refining the images on the paper plotter, next previewing the movie with FRED, and finally moving to the microfilm recorder to complete the film.

When requested, ZAPP will output images to two devices on a single run. For instance, use of the microfilm as a primary device and the paper plotter as a secondary device provides a hard copy record for debugging, documentation, and discussion. Such output is usually done in "cartoon mode", with successive miniature images arrayed in vertical columns on the paper.

For such use, only key frames are desired. This also applies to many test runs on the microfilm recorder. ZAPP allows the user to specify <u>frame output filters</u> which determine which frames are output to the primary device and to the secondary device. For example, she can request specific individual frames, or every nth frame beginning at a certain one, or all "essential" frames, where an essential frame is defined as one in which an agenda action starts or stops.

C. Aids for picture management and timing control

The ZAPP animator assigns symbolic names to her pictures and transacts with the system on those terms. ZAPP manages space for these pictures and does dynamic storage allocation. Each picture has a "visibility" bit, and with it the user can temporarily "hide" images. In addition, there is a mechanism whereby one can apply a transformation to all pictures currently defined.

Because ZAPP executes in a batch environment, and the content of a picture is generally fixed, ZAPP represents each picture in a single block of storage. Pictures, however, can be changed, and the system generates new blocks appropriately. Two pictures can be combined, individual curves and points within pictures can be interrogated or modified, and new curves and points can be added.

ZAPP has an absolute frame counter, and also a relative frame counter that can be reset to 0 at the beginning of each scene. All timing references are interpreted with respect to the relative counter. The user can decide if time is to be clocked in seconds or frames, and also determine the ultimate correspondence between seconds of movie time and frames produced by the microfilm recorder.

IV. Results and Conclusions

In the past 18 months, ZAPP has been used in the following projects:

-- to produce one of a series of educational films on relativity theory (8);

-- to make a short art film (9);

-- to produce animated representations of the output of a simulation model of a subway station(10);

-- to develop visualization techniques for presenting a large body of accumulated data on skull growth (11);

-- to investigate transformations of type fonts considered as images (3);

-- to develop two prototype program animation systems, which produce animated representations of the execution of computer programs (12); and,

-- to develop a program illustration system that produces documentation consisting of program text interspersed with images depicting program state (13).

The original design goals have been met, and, despite OS/370, ZAPP has survived. Yet we have often regretted the multi-lingual constraint, which led us to implement in FORTRAN and assembler. This doubled the implementation cost, made ZAPP too unreliable to transport, and allowed the uglinesses of FORTRAN to impair the readability and make awkward ZAPP programs written in any of the host languages. It would have been better to implement in a structured assembly language or systems implementation language, even if ZAPP had then been inaccessible from FORTRAN or PL/I. Another major difficulty stems from the awkwardness of a multiple machine environment without adequate data communications linking the machines. We are therefore beginning work with a medium-scale minicomputer which should ultimately support all our animation activities, both demonstrative and algorithmic.

For a more complete description the reader is referred to References 14.

Acknowledgements

Thanks to Margie Guerin for implementing and then maintaining ZAPP; Tom Britton, Eric Onasick, Tom Horsley, Mike Tilson, and Greg Hill for suggesting and adding features; and the National Research Council for financial support.

References

(1) Marjorie Guerin, A System for Computer Animated Film Production in a Batch Processing Environment, M.Sc. Thesis, Dept. of Computer Science, University of Toronto, 1973.

(2) Michael D. Tilson, Editing Computer Animated Film, M. Sc. Thesis, Dept. of Computer Science, University of Toronto, 1975.

(3) Ronald M. Baecker, Stan Bevington, and Thomas Britton, "Artistic Computer Transformations of Two-Dimensional Text and Images, Final Report to the Canada Council Explorations Program, 1975.

(4) Ronald M. Baecker, "Picture-Driven Animation", Proceedings of the 1969 Spring Joint Computer Conference, 273-288.

Ronald M. Baecker, Interactive Computer Mediated-Animation, MIT Project MAC-TR-61, 1969.

(5) Leslie Mezei and Arthur Zivian, "ARTA, An Interactive Animation System", Proceedings of the 1971

International Federation of Information Processing Societies Conference, 429-434.

Nestor Burtnyk and Marcelli Wein, "Computer Generated Key-Frame Animation", in *Computer Animation*, John Halas (Editor), Hastings House, 1974, 39-44.

(6) Peter Foldes, "La Faim (Hunger)", Color Sound Film, The National Film Board of Canada, 1974.

(7) Ronald M. Baecker, "GENESYS: Interactive Computer-Mediated Animation", in *Computer Animation*, John Halas (Editor), Hastings House, 1974, 97-115.

Ronald M. Baecker, Lynn D. Smith, and Eric Martin, "GENESYS: An Interactive Computer-Mediated Animation System", Color Sound Film, MIT Lincoln Laboratory, Lexington Massachusetts, 1970.

(8) Bruce and Kathryn Cornwell, "Doppler Effect and the Twin Paradox", Color Film, Houghton-Miflin Company 1974.

(8) Eric Onasick, "ZAPPEX", Color Film, Dynamic Graphics Project, Computer Systems Research Group, University of Toronto, 1974.

(10) Ronald M. Baecker and Thomas Horsley, "Computer-Animated Simulation Models: A Tool for Transportation Planning", *Transportation Research Record*, to appear.

(11) Ronald M. Baecker, Eric Onasick, and Gordon Thompson, "Skull Growth" Films, Dynamic Graphics Project and Faculty of Dentistry, University of Toronto, under production.

(12) Ronald M. Baecker, Producer, "Program Animation Film Sampler", Dynamic Graphics Project, 1974.

Ronald M. Baecker, "Two Systems Which Produce Animated Representations of the Execution of Computer Programs", ACM *SIGCSE Bulletin*, Volume 7, Number 1, 158-167.

(13) Edward Yarwood, Toward Program Illustration, M.Sc. Thesis, Dept. of Computer Science, University of Toronto, 1974.

(14) Ronald M. Baecker, Marjorie Guerin, Michael D. Tilson, "A Computer Animation Facility for Research and Educational Filmmaking", currently being reviewed.

Marjorie Guerin, ZAPP Version 2.1 User Manual, Dynamic Graphics Project, November 1974.