# ANIMATION OF CONTINUOUS COMPUTER SIMULATIONS

C.M. Woodside and Richard Mallet

Computer Center, Carleton University

## ABSTRACT

A block-oriented graphics program called ANIM8 has been developed for animating the results of continuous simulations. The time-tracks of relevant variables in the simulation are fed into the ANIM8 program and drive the positions, rotations, scalings and other variables in the visual display. For example, a simulation of vibrating system could be represented via a sketch of the system, with the parts moving.

It is presently implemented on a central timeshared computer with a slow online graphics facility (a Data-disc television-oriented facility), with which the picture elements can be defined and the graphic output structure debugged in advance of simulation. The simulation is performed and its output is passed as data to ANIM8. Running ANIM8 interactively the user can step through the display, explore detailed time sequents frame by frame, or film an animation on 8 mm film.

The block structure of ANIM8 appears to be convenient, and sufficiently flexible and powerful. Although it has obvious limitations over a statement-oriented language, it has certain efficiencies also. The blocks are operators which operate on subpictures (rotate, scale, move, join it to other subpictures, distort it, etc.).

The paper will elaborate the structure of ANIM8 and give an example of an animated simulation, including filmed animation.

## ABRÉGÉ

On a développé un programme, orienté vers les blocs pour la représentation graphique, appelé ANIM8 pour l'animation des résultats des simulations continues. Les trajectoires en fonction du temps des variables appropriées de la simulation sont injectées au programme ANIM8 et elles commandent les mouvements, les rotations, les changements d'échelle et les autres variables de l'image visuelle. Par exemple, on peut représenter la simulation d'un système oscillant par un dessin du système dont certaines parties sont en mouvement.

ABRÉGÉ (cont)

À présent le système fonctionne dans un ordinateur central à temps partagé avec un équipement graphique connecté à action lente (Data-disc avec orientation vers la télévision) sur lequel on peut définir les éléments de la représentation et vérifier la structure du rendement graphique avant que la simulation soit faite. Puis, la simulation faite, on passe les résultats au programme ANIM8 comme données. Utilisant ANIM8 dans la méthode interactive, l'utilisateur peut faire voir les images en séquences, explorer la séquence d'images une par une ou filmer avec une caméra de 8 mm.

La structure en blocs d'ANIM8 semble être commode et, en même temps, assez souple et puissante. Quoiqu'il y ait des limitations évidentes par comparaison avec un langage d'instructions, il y a aussi des avantages intéressants. Les blocs sont des opérateurs qui manient les sous-images, les font tourner, changer d'échelle, se déplacer, les faire se joindre a d'autres sous-images, les déformer, etc.

On donnera des détails de la structure d'ANIM8 et aussi on représentera un exemple d'une simulation animée, y compris un film animé.

# ANIMATION OF CONTINUOUS COMPUTER SIMULATIONS WITH ANIM8

C. M. Woodside & R. Mallett

Carleton University, Ottawa

ANIM8 is a program which constructs a depiction of a dynamic system in continuous motion, frame by frame for recording on motion picture film. The motion is defined in advance by running a separate simulation of the dynamics of the system and storing the trajectories of key variables over time. ANIM8 is therefore a graphics "language" driven by time-varying parameters. A block structure which is believed to be unique was chosen for ANIM8, and is the major content of this paper.

Block structures for programming must be compared to language structures. In general, progress has been from block structures (CSMP, GPSS) towards language structures (CSMP-III, SIMSCRIPT) and the latter have much more flexibility -- for instance in the use of subroutines, of loops, and of arrays. On the other hand block structures have (I believe) a lower learning threshold and are a natural mode of expression for people who think graphically -- which may be particularly useful in graphics. These factors may be more important in a university setting, where many new users have a transient contact with a system, than in some others.

## The Type of Animations Considered

The animation of technical and scientific simulations is the underlying purpose of ANIM8. The pictures will therefore be mostly simple and abstract, and probably not formed by sketching. The pictures are made up of subpictures, transformed and combined, and it has been assumed so far that the dynamic changes occurring in the animation can be represented by changes in the transformations and combinations rather than in the definitions of the basic subpictures. This assumption is restrictive, but can be relaxed by simple extensions to the processor.

An example of a simple technical animation is the motion of the "tower" shown in Figure 1 when forced into nonlinear oscillation by external pulses of force, for instance by a pulsed jet of air. The magnitude of the force is indicated pictorially by the position of a flap in front of the blower symbol, and the three-dimensional tower is shown in a slant projection. The motion of the tower is completely determined by its angle to the vertical, $\theta$ in Figure 2; however to represent the rotation about two different axes the vertical and lateral displacements of the centre of the base are also used in the specification of its position. All these values are calculated in the simulation.

## The ANIM8 Block Structure

The example above will be used to illustrate the specification of an animation. The underlying philosophy of the block diagram is that pictures (or subpictures) are stored by certain types of blocks; they flow out along arrows to blocks which successively transform them by rotation, scaling, and translation in three dimensions, and to blocks which combine subpictures in various ways. The pictures are arrays defining three-dimensional outlines (no hidden lines) and the transformations are applied to these arrays point by point. Figure 3 shows a PIC block which stores the basic shape of the tower, a ROT block to rotate it by θ about the z-axis, and a SHF block to apply the x and y translations.

The rotation by the ROT block is actually controlled by two sets of angles; a fixed parameter vector ($p_1$, $p_2$, $p_3$) attached to the block and a variable vector ($\theta_1$, $\theta_2$, $\theta_3$) applied as input number 2 to the block (the sub-picture array itself is connected as input number 1). The variable vector comes from a block which reads the simulation outputs from a file. One set of parameters from the simulation must be supplied for each frame. The shift block is handled similarly. For simple motions and for debugging purposes a simple time function (with 3 components) can be tabulated and used to drive the time varying parameters without requiring a data file from a simulation.

An ANIM8 animation is defined and run exactly like an analog computer simulation; in fact ANIM8 was developed by modifying a simulator (SIMUL8) which imitates an analog computer. We have had great success with SIMUL8, and this was one of the reasons for proceeding with the block structure. The "variables" are all either sub-pictures (2 dimensional arrays) or parameter vectors. The blocks are all numbered consecutively and the connections are specified by stating which blocks are connected to input 1 and input 2 of each block. Alterations are made by "repatching". If the output of a block is a subpicture array then that block can be plotted, either at a specified time or over a sequence of times; a number of blocks can be plotted simultaneously in each frame.

The complete animation diagram for the example is shown in Figure 4, and the numerical specification in Figure 5. The variable parameters are defined in the Figure by tabulated sequences internal to ANIM8; a minor change gives the connection to external data from a simulation.

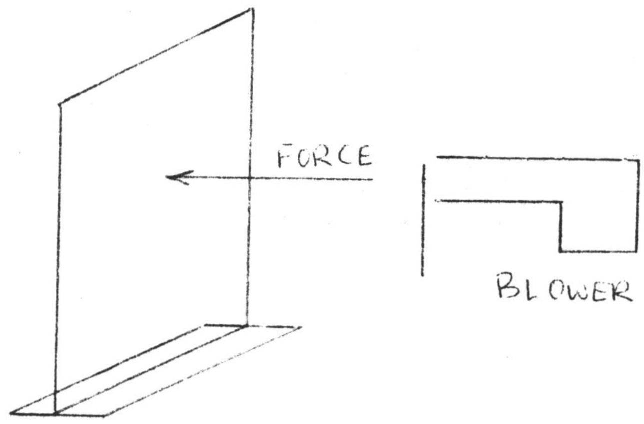


Figure 1 "Tower" Example

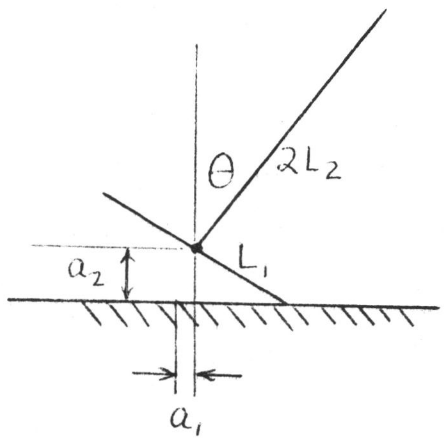


Figure 2 Schematic

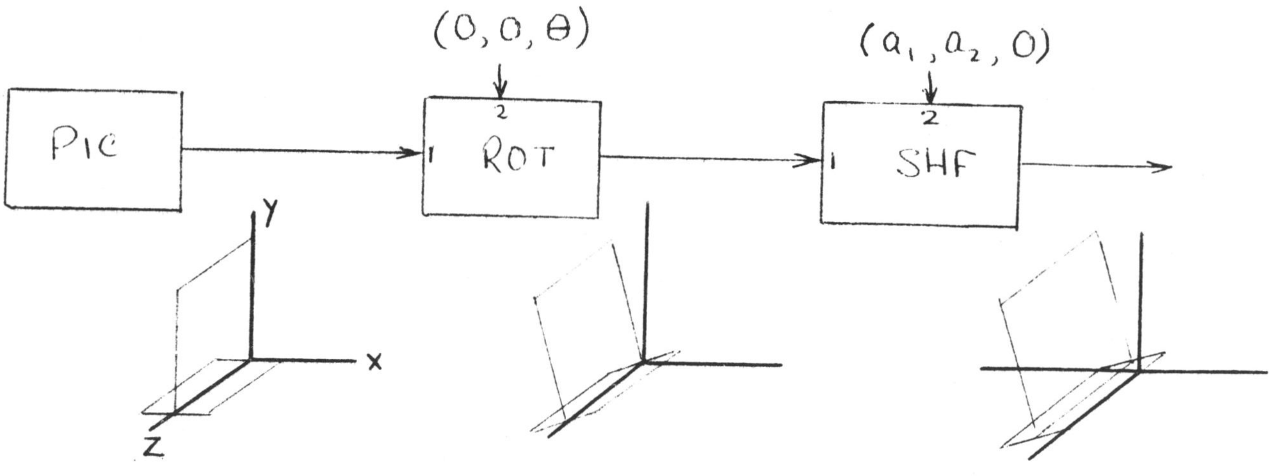


Figure 3 Transformation with Parameters

The block types implemented in the first version on ANIM8 are:

- PIC : Store a defined subpicture.
- ROT, SCA, SHF : Rotate, scale, and shift (as discussed above).
- SEQ : Store a defined time sequence in three parameters.
- TFN : Connect a parameter vector to a time sequence defined on an external file.
- PIN : Connect two subpictures together.

We plan to try to use this implementation for a while before expanding the choice of operations; it is not yet decided that the block structure merits further developement relative to the alternatives.

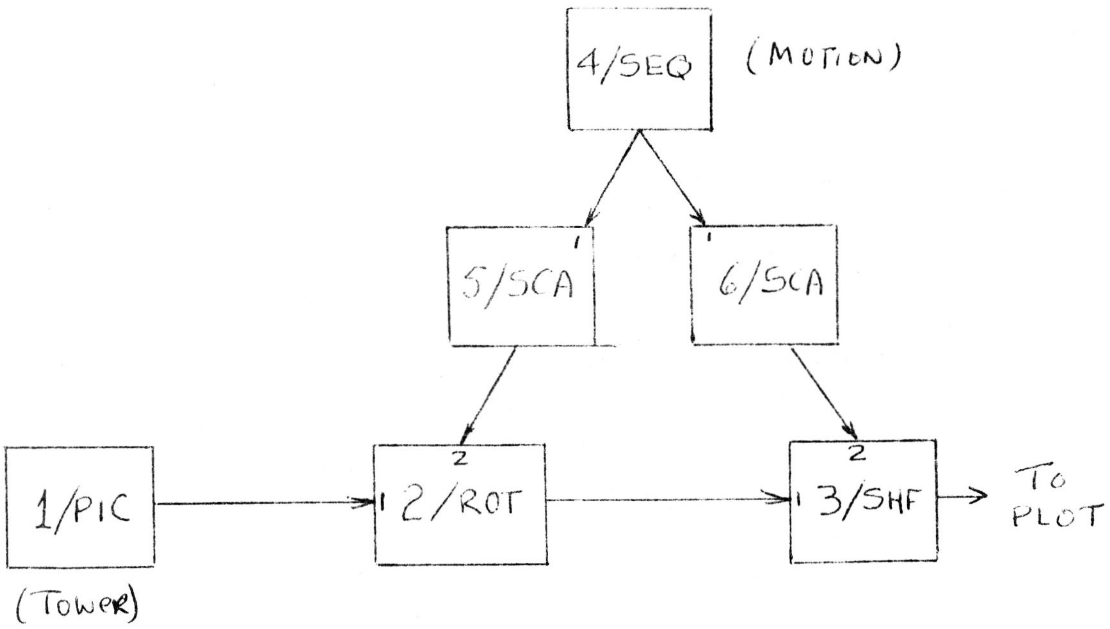


Figure 4: Animation Diagram for the Moving Tower

## Block Definitions

| Block | Type | Inputs 1 | Inputs 2 | Inputs 3 | Parameters 1 | Parameters 2 | Parameters 3 |
|---|---|---|---|---|---|---|---|
| 1 | PIC | 1 (for Table 1) | | | 0 | 0 | 0 |
| 2 | ROT | 1 | 5 | | 0 | 0 | 0 |
| 3 | SHF | 2 | 6 | | 0 | 0 | 0 |
| 4 | SEQ | 2 (for Table 2) | | | 0 | 0 | 0 |
| 5 | SCA | 4 | | | 0 | 0 | 1 |
| 6 | SCA | 4 | | | 1 | 1 | 0 |

## Table Definitions

Table 1 (picture)

| pen | x | y | z |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 3 |
| 0 | -1 | 0 | 3 |
| 0 | -1 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 8 | 0 |
| 0 | 0 | 8 | 3 |
| 0 | 0 | 0 | 3 |
| 0 | 0 | 0 | 0 |

(10 points)

Table 2 (time sequences)

| t | $a_1$ | $a_2$ | $\theta$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 3 | .1 | .4 | 20 |
| 6 | 0 | 0 | 0 |
| 9 | -.1 | .4 | -20 |
| 12 | 0 | 0 | 0 |

(5 points)

Figure 5: Numerical Definition of the Animation in Figure 4