



A COLOR ANIMATION SYSTEM

BASED ON THE MULTIPLANE TECHNIQUE

Marc Levoy

Program of Computer Graphics
Cornell University

Abstract

This paper describes an animation package currently under development at the Cornell Program of Computer Graphics.

The basic algorithm employed is linear or non-linear interpolation between successive pairs of key frames. These key frames are composed of artwork input by the animator on a graphic tablet and displayed on either a black and white vector scope or a color halftone CRT. The initial working environment is two-dimensional, and the individual images are combined using a multiplane cel animation technique to produce depth and motion illusions. Real-time film previewing, utilizing an on-the-fly interpolation algorithm, provides the artist with instant playback of animated sequences.

KEYWORDS: Key-frame Animation, Real-time Animation, Multiplane Animation, Interactive Computer Graphics
COMPUTING REVIEWS CLASSIFICATION: 3.41, 8.2

I. Introduction

The last five years have seen the development of several sophisticated computer animation systems. To review a fairly well established taxonomy, two general categories can be identified: language-driven systems and picture-driven systems. Each type offers certain advantages to the artist, although each contains certain inevitable drawbacks. These may be summarized briefly as follows:

Language-driven systems:

Beginning with Kenneth Knowlton's pioneer pixellation films of the early sixties and continuing up through the current research of the Ohio State University Group (2) and others, all language driven animators have in common a desire to achieve maximum flexibility in their animation environment through the provision of a wide repertoire of picture manipulation commands. By allowing the artist to specify any two-dimensional or three-dimensional transformation upon geometries in the working data space, these systems provide considerable freedom for the development of artistic scenarios.

There are, however, distinct difficulties in this approach. First the generation of three-dimensional objects with sufficient ease and of adequate complexity to serve as artistic elements in computer animation is a problem of considerable complexity in itself. Second, since the working environment of these animation systems is manipulated and distorted according to commands entered on a computer terminal, the artist must acquire at least some familiarity with language syntax.

Picture-driven, key-frame animation systems:

In systems of this class the animator prepares selected images at key intervals in the sequence and sketches them on a graphic tablet.

Interaction between the artist and his animation is purely pictorial, and picture generation proceeds from these key drawings instead of from programmed commands. By subdividing each sketch into discrete strokes, and from there into a large number of closely spaced points, the computer can be programmed to generate frames intermediate to the key drawings using a simple linear interpolation algorithm.

Several existing animation systems are based on this approach, including those developed by Wein (8) and Catmull (1). There are, however, major problems with key-frame animation. First, there must be a one-to-one correspondence of strokes between the two terminal sketches in a transformation. While this scheme provides a simple means for controlling the overall transformation, certain classes of actions become extremely difficult to achieve, such as large rotations of objects. Second, linear interpolation between two substantially disparate drawings invariably results in the "collapsing phenomenon", wherein the lines of the drawing appear to disintegrate and mingle chaotically during the course of the transformation. Wein has provided several excellent suggestions for alleviating these artifacts of key-frame interpolation (7). A third impediment relates to the organization of timings in the key-frame method. By adopting an algorithm for animation wherein each transformation is planned separately, the problem of maintaining smooth transitions becomes critical. In most of the work produced to date the tempo of the key frames is highly apparent, and since continuity is essential, the key-frame method presents a definite constraint in this area. Finally, a limitation is imposed by forcing the artist to inhabit an essentially two-dimensional world. While it is a simple matter of draftsmanship to produce a static image which successfully portrays depth, it is entirely a different problem to produce key-frame interpolations of objects that retain true perspective throughout the transformation.

This work has been partially sponsored by the National Science Foundation under grant number DCR74-14694 entitled "Development of Computer Graphics Techniques & Applications."

The Cornell Program of Computer Graphics is currently developing an animation system for eventual use in a production studio environment. The system is basically designed on the key-frame model of computer animation, for which excellent descriptions can be found in (4) and (8). It also provides three-dimensional capabilities, but, in order to prevent these additional degrees of freedom from becoming too unwieldy, they are dispensed only in a carefully controlled manner, as described in the next section.

II. THE CONCEPT OF MULTIPLANE ANIMATION

In the animated cartoons of the 1920's, the backgrounds were designed strictly as static, two-dimensional backdrops. They lent a sense of place and scale to the animated characters in the foreground cel, but rarely dealt with the space between these figures and their horizon.

When Walt Disney began work in 1936 on his first feature-length film, Snow White, it became necessary to depict settings wherein the characters were separated by vast distances from their backdrops. Carefully constructed perspectives could provide an impression of depth as long as the observer's viewpoint was static, but a very difficult problem arose when the script called for a zoom or a panoramic sweep. In order to simulate a movement of the imaginary observer toward a character in the foreground, the figure should enlarge in the field of view. This can be accomplished simply by bringing the camera closer to the animation platten. The background scene also increases in apparent size during the zoom, but at a much slower rate due to its greater distance from the observer. If the background cel were kept flush with the foreground figure as it had always been in traditional animation, it would enlarge as rapidly as the foreground and the illusion of depth would be lost.

This problem was solved by Disney with the aid of an awesome machine called a multiplane camera, shown in Figure 1a. Each layer of animation is situated at a different distance from the camera and illuminated independently. Thus, to achieve a realistic zoom, each cel layer is moved toward the camera at a speed inversely pro-



Figure 1a: The multiplane camera stand used by Walt Disney in his productions of the 1930's.

portional to its distance from the imaginary observer. The differential ratios in these rates determine the apparent position of each drawing plane in depth. This was and still is one of the cleverest devices ever to be used in animation. Unfortunately, each cel layer required several technicians to operate it, and for this reason it has been used sparingly since the first feature films. Nevertheless, it represents an excellent algorithm for generating animated sequences of correct perspective and realistic three-dimensional motion from a set of essentially two-dimensional drawings (3).

III. THE CORNELL ANIMATION SYSTEM

1. General organization:

Figure 2 illustrates the layout of the animation package. It has been developed as a dual-display system, alternating between a black and white vector scope and a color half-tone CRT.

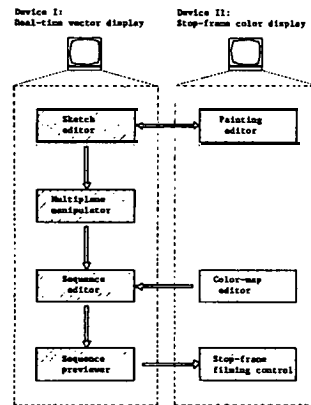


Figure 2: Schematic organization of the Cornell animation system, showing the interface between the vector and color graphic systems. For a complete description of the equipment configuration of the Cornell laboratory, refer to (5), also in this issue.

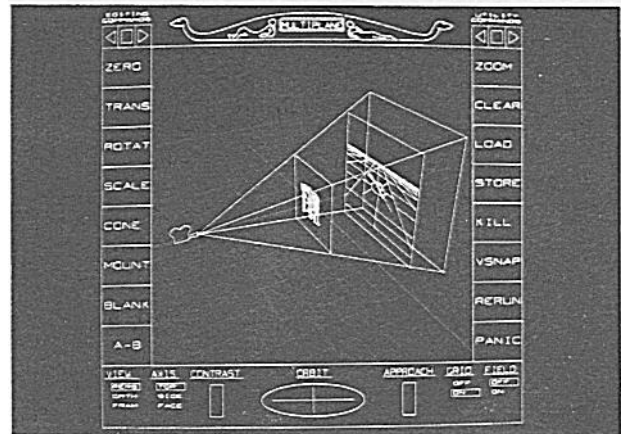


Figure 1b: The multiplane editor of the Cornell animation system (with its menu).

Input of the artwork can be initiated from either device, depending upon the graphic medium selected by the artist. This process is entirely pictorial, and the primary input means is a graphic tablet and pen.

The data then passes to the multiplane manipulation routines. Here, the sketches and paintings are combined into a multiplane animation environment analogous in concept to that used by Walt Disney. This segment of the animation package operates exclusively on the vector display, since that device offers the best real-time capabilities.

At this stage, the data consists of a series of key frames specifying the exact state of the animation at selected intervals in the sequence. The final step is for the computer to perform the mathematical interpolation between these key frames, and to display the resulting sequence on one of the output devices. Two means for previewing are available to the artist, real-time black & white playback or stop-frame color playback. The restriction of real-time playback to the vector scope is due entirely to the limitations of available technology. After examining the animation, the artist then has the option to refine either his artwork or the multiplane setup and then repeat the procedure. When the previewing finally yields satisfactory results, production filming may begin. For this purpose, each composite image is displayed one frame at a time on the color CRT and filmed using a permanently mounted 16mm camera. This process, although slow, is performed automatically by the computer and requires no user intervention.

In the following sections, each of the three basic steps outlined above will be explained in detail, and examples will be provided for each stage of the process.

2. Building the two-dimensional planes:

Each plane in the multiplane environment contains one animation sequence developed by the artist. Several different animation techniques are provided, but all of them generate a continuum of two-dimensional images for that plane.

a) Key-frame interpolation of line drawings with linear output (Figure 3a)

The procedure for developing a key-frame sequence from line drawings is well known and has been described elsewhere (8). The final product of this method is a sequence of frames containing vector approximations of interpolated curves. This may be displayed on either a vector scope or a raster CRT.

b) Key-frame interpolation of line drawings with color output (Figure 3b).

Utilizing a simple area-filling algorithm, contiguous areas enclosed by the original line drawing can be flooded with colors from a "palette" and displayed on a color raster CRT, yielding an image similar in appearance to a traditionally opaqued animation cel. If a set of "seeds" to drive the

area-filler is interpolated along with the profile lines, intermediate frames can be automatically generated with full color opaquing. When these frames are later superimposed over other sequences in the multiplane environment, multiple-cel opaquing is effected.

c) Static color backdrop (Figure 4a)

The rear planes in Disney's multiplane animations contained highly elaborate backgrounds, far too complicated to be viably animated. Many of them were oil paintings on glass plates, a nearly incredible combination of materials to handle, but one which allowed great variation in texture and tonal delicacy. As part of the Cornell animation system, a painting program has been developed in which the artist works with computer-generated analogies of traditional drawing implements, such as pens, brushes and washes. (This program is similar in design to the SHAZAM system of Xerox and its descendants in other graphics laboratories (1) (6). Using this program, full color backdrops of considerable complexity can be achieved.

d) Color-palette interpolation of static backdrops (Figures 5a & 5b)

Although the complexity of painted backgrounds prohibits geometrical animation, there exists equally interesting possibilities in the interpolation of the color palette. For example, suppose that the artist colors a rendering of an imaginary temple in its nighttime array of blues and deep purples. Then, without repainting the basic figure, a sequence of palettes can be specified that depict the play of light and color in the hall as the sun rises. These palettes can finally be used by the computer as key frames from which an entire animated sequence of subtle chromatic shifts can be produced.

e) Back-projection of three-dimensional simulation (Figures 6a & 6b)

Constraining the artist to inhabit an exclusively three-dimensional world has definite drawbacks. There are, however, instances in which three-dimensional simulation promises to yield the best backgrounds for a particular scenario. Architectural settings exemplify stagesets that would not alter in shape, yet might be viewed from a variety of different angles during the course of an animation. The Cornell animation system provides these three-dimensional simulation capabilities, although only in the following carefully structured manner. Attaining the correct arrangement of two-dimensional animation planes within a three-dimensional stageset is analogous to the problem of combining animation and live-action filming in one sequence. Unless the spatial arrangements are executed flawlessly, one of the media will "go flat", or seem to lose its perspective depth. Figure 6a illustrates this difficulty. The simultaneous juggling of the two image types with respect to perspective alignment would be extremely cumbersome to handle in this setup. One solution to this problem, and the one adopted for the Cornell animation system, is to separate the two media, compose one independently, then superimpose the other at a later stage. Specifically, the three-dimensional simulation is worked out

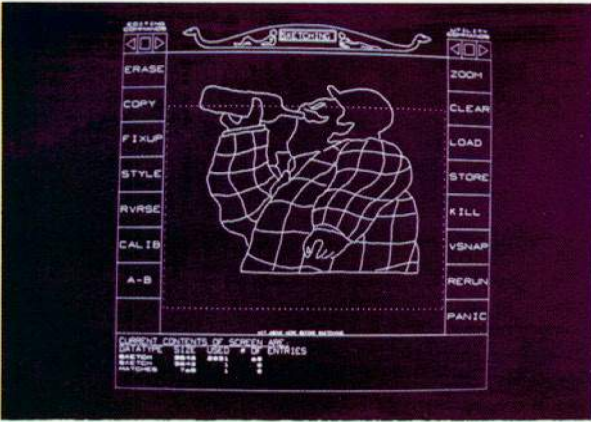


Figure 3a: Key-frame interpolation of line drawing - vector output



Figure 3b: Key-frame interpolation of line drawing - color output



Figure 4a: Static color background displayed on color screen.



Figure 4b: Vector approximation of 4a for display on vector scope (allowing real-time manipulation).

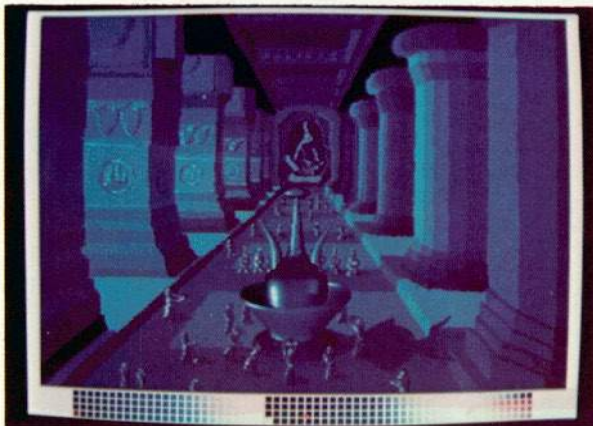


Figure 5a: Combination of two-dimensional and three-dimensional animation, in night colors.

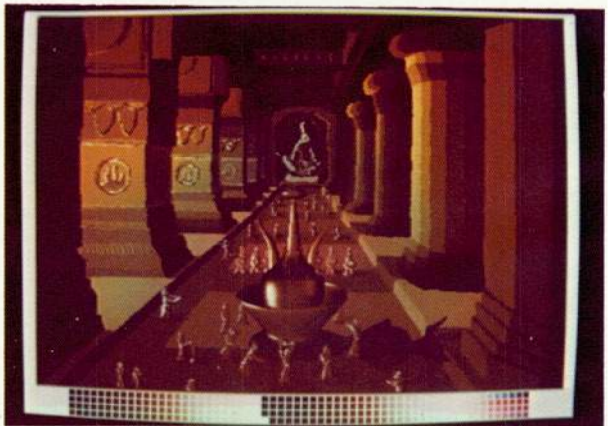


Figure 5b: Combination of two-dimensional and three-dimensional animation, in morning colors.

Figure 6a: Direct combination of 3d and 2d animation

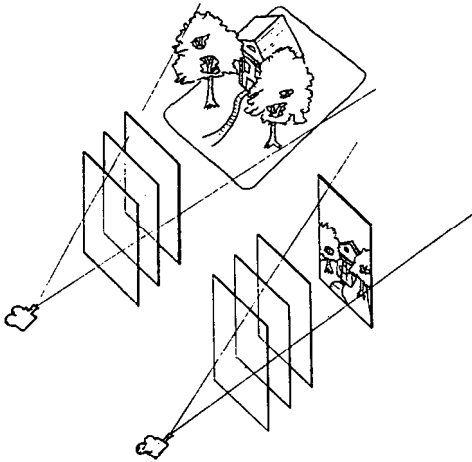


Figure 6b: Back projected combination of 3d and 2d animation.

first, complete with camera angles and motion dynamics. Then the resulting sequence is back-projected as a continuum of two-dimensional images on the background plane of the multiplane environment, as diagrammed in Figure 6b. Finally, the artist proceeds to develop his two-dimensional animation as a superimposition over the simulation sequence. In much the same way that traditional animation was executed to match a pre-recorded sound track, the multiplane animation in this example would be altered and refined to overlay a previously established background sequence. using this method, correct perspective and alignment between the two media are virtually assured. The fountain and temple in Figures 5a & 5b were combined in this manner, with the temple input as a two-dimensional painting and the fountain as a three-dimensional simulation.

3. Constructing the multiplane animation:

Figure 1b illustrates the geometrical structure of the Cornell animator (compare to Figure 1a). The imaginary observer in this system defines a three-dimensional cone or frustrum of vision which is transpierced at carefully defined intervals with data planes. The two-dimensional raw animation sequences of the artist reside on these planes.

By manipulating the object planes and observer's position, the artist can choose the best views and sequences to tell his story. As Figures 7a & 7b illustrate, the multiplane effects are achieved by translating the observer's position in any of the three directions or by tilting the line of sight in any of the three degrees of rotational freedom. In addition, translations, rotations and scaling functions may be specified for each plane in the environment for the possibility of interrelational movements among the cels of the composite.

For the previous stages, some of the techniques described utilize a vector-drawing display while others depend on color raster graphics. For the next stage in the process however, all work must be representable on the vector scope. To provide this capability, an edge-finding algorithm can be utilized to generate a coarse vector approximation of those drawings created originally for the color display. This vector simulcrum, an example of which is shown in Figure 4b, contains substantially less information about detail and texture than its color genesis, but it is adequate for the task of arranging the planes in space and specifying the sequence dynamics.

Interaction between the artist and the computer is entirely pictorial, requiring no knowledge of programming or language syntax. Motion feedback is immediate, in real-time and at full resolution, utilizing the fast vector drawing system. Throughout this process, the locations of the planes and observer are being constantly tracked, so that acceptable sequences can be replayed automatically and included in the final animation.

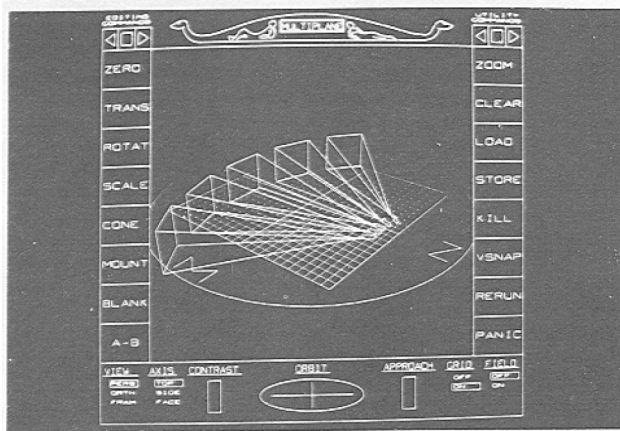


Figure 7a: Manipulating the observer's frustrum of vision in the multiplane editor.

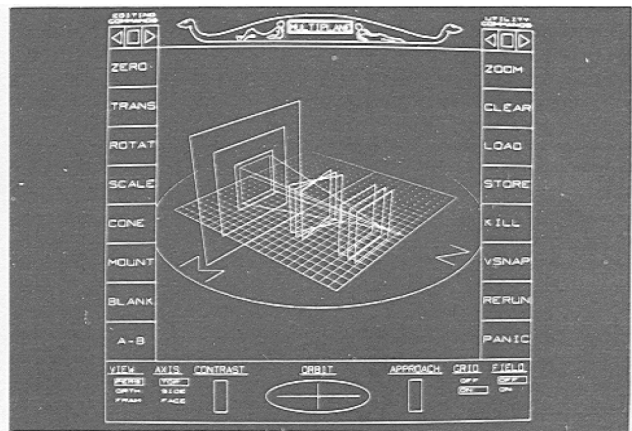


Figure 7b: Manipulating the object planes in the multiplane editor.

4. Real-time previewing:

One important feature in any animation system is the ability to preview animation sequences at cine projection speed (24 frames per second), or at least at standard animation speed (12 frames per second). In the Cornell system, real-time playback is achieved using the fast vector drawing display. For this purpose, the vector approximations of all color drawing is utilized.

A hardware matrix processor is capable of handling in real-time whatever translations and rotations may be specified for each plane in the environment. Unfortunately, however, the linear interpolations utilized in the key-frame sequences require substantial processing time since the calculations must be applied in software to each of the thousands of points in each sketch. These massive computations presently make real-time playback extremely difficult to achieve.

The possible programming strategies for achieving real-time playback of interpolated sequences fall into one of two categories:

- 1) Pre-calculated interpolation, in which the geometry of each intermediate frame is computed before the previewing begins and deposited on a large capacity, high speed mass storage device for subsequent playback.
- 2) On-the-fly interpolation, in which the geometry of each intermediate frame is computed by the host computer just prior to its display on the screen.

The inherent disadvantages of pre-calculating the in-between frames and storing them on the disk are fairly evident. First, it incurs waiting time between the input and playback phases of the process. Second, it places an absolute limit on the number of frames that may be previewed at once, depending upon the particular storage medium. Third, any subsequent modifications to the animation require that the entire sequence be recomputed before it may be previewed again.

By using an intelligent graphics processor that is substantially independent from its host computer, sufficient computing power remains in the CPU to implement an on-the-fly interpolation scheme. Furthermore, utilizing algorithm and code optimization techniques, the computations necessary to perform simple linear interpolation can be diminished significantly from the basic formula.

Three processes are actually involved in producing the continuous animation on the screen:

- 1) Loading of each pair of key frames from the disk.
- 2) Interpolation of the sketch data.
- 3) Matrix transformation and display of the intermediate frames.

The key to producing this fast continuous animation lies in the fact that each process is achieved by a separate part of the computer system, operating in an asynchronous, multi-tasking environment. The three independent devices involved are the mass storage controller, the CPU, and the display processor.

These three asynchronously operating devices are slaved to the previewer in the Cornell system using a network of pseudo-simultaneous co-routines executing in the central computer, as shown in Figure 8. At the start of key frame (n), the disk controller is instructed to begin transferring the data for key frame(n+1) to a core buffer. At the same time that the disk transfer is triggered, interpolation commences on the first plane of key frame (n). When this plane is complete, the display processor is commanded to transform and display the plane. This is achieved by loading the plane's geometry into the display pipeline along with the matrices desired for translation and rotation. Meanwhile, the CPU is working on the interpolations for the second plane. When all planes have been computed and processed for display, work begins again for the next frame. By the time the entire key frame is exhausted, the disk operation is complete and the next key frame is ready for interpolation.

Since the display processing operates faster than the interpolations, the critical path includes all of the interpolations, but no other processes, as the diagram indicates. If by chance the work required to display a single frame is completed ahead of schedule, an internal clock delays the program for the proper interval of time. The work will never fall behind schedule, since before the playback process begins, each sketch has been subdivided only into a number of points that can be handled at 12 frames per second. For our present equipment configuration, this limit has been measured at about 2800 points, which allows sketches of far greater complexity than can be previewed using the pre-calculation method, due to disk-to-core transfer speed limitations. Of course, sketches of considerable complexity take on a slightly coarser appearance due to this filtering/subdivision algorithm, but the results so far have been very satisfactory. For sketches containing an initial point count of under 5000, the filter-down is almost impossible to detect. Very seldom has any sketch produced by the artists working on the Cornell system exceeded 7000 points, which makes the technique viable for use in a production environment.

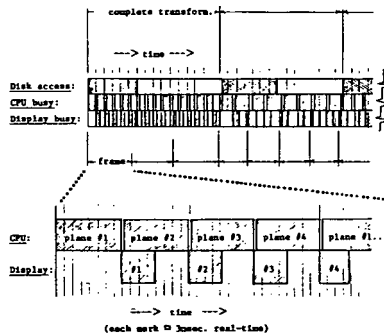


Figure 8: Time-resource diagram of the real-time playback system, showing the critical path and the multi-tasking.

In addition to the real-time playback capability, individual frames in the animated sequence can be transferred to the color display for stop-frame examination. The image thus presented to the artist looks exactly as will the final film, including multiple-cel opaquing and full color shading.

IV. CONCLUSION

This paper has presented an overview of the key-frame computer animation system currently under development at the Cornell Program of Computer Graphics.

The basic algorithm employed is a linear interpolation between successive pairs of key frames. These key frames are composed of artwork input by the animator on a graphic tablet and displayed on either a black and white vector scope or a color halftone CRT. The attributes of this animation package are:

- 1) The initial working environment is two-dimensional, and any of the following graphic media may be used:
 - a. Free-form line sketches.
 - b. Line sketches with color infilling.
 - c. Color paintings.
 - d. Two-dimensional projections of complex three-dimensional scenes.
- 2) These two-dimensional images are combined using a multiplane cel animation technique to produce depth and motion illusions.
- 3) Real-time film previewing, utilizing an on-the-fly interpolation algorithm, provides the artist with instant playback of animated sequences.
- 4) All interaction with the program is pictorial in order to facilitate the use of this system by the non-programming artist.

It is evident that numerous approaches have been taken to computer animation. This can be ascribed, to a large extent, to variations in artistic taste. There are, however, problems common to all animation systems which must be solved before they can be viably used for production work. These involve increasing the resolution, allowing for real time playback at this increased resolution, and providing a more efficient input system. Future research in computer animation must address these issues, as described below:

- 1) Most currently available color frame buffers are designed for the 512 x 512 nominal resolution of a video monitor. In order to obtain a satisfactory image for theater projection, a resolution of at least 2000 x 2000 is required. The enormous data storage problems this implies, and the lack of any viable device for previewing such images, pose formidable problems for computer animation systems.

- 2) The complex pixel-by-pixel computations required to generate a single frame of color animation make real-time color previewing very difficult to achieve. The only viable solution yet evolved for this limitation is to compute each frame beforehand and store it on a video disk or stop-frame video tape recorder for subsequent playback in real-time.
- 3) Most animation systems currently require that the artist input his sketches manually, using a graphic tablet and pen. This time-consuming extra step can be eliminated through the use of a high-precision optical scanner. The edge-finding and stroke identification algorithms necessary for such a procedure pose complex problems in themselves, but if solved, promise to further ease the labor burden on the animator.

V. REFERENCES

1. Catmull, Edward, (Personal conference at New York Institute of Technology), New York, (1976).
2. Defanti, Thomas, "The Graphics Symbiosis System", Real-Time Film Animation, annual report to the NSF, submitted by the Computer Graphics Research Group at Ohio State University, (1973), pp. 5-116.
3. Finch, Christopher, The Art of Walt Disney, Walt Disney Productions, (1975).
4. Gracer and Blasgen, "Karma: A System for Storyboard Animation", UAIDE, (1970), pp.210-224.
5. Greenberg, Donald P., "An Interdisciplinary Laboratory for Graphics Research and Applications", Proceedings of the Fourth Annual Conference on Computer Graphics, Interactive Techniques, and Image Processing (July, 1977).
6. Shoup, Richard, (Presentation of the SHAZAM animation system at the Third Annual SIGGRAPH Conference), Phila. Pa., (1976).
7. Wein, Marcell, "Interactive Skeleton Techniques for Enhanced Motion Dynamics in Key-Frame Animation", Communications of the ACM, Oct. 1976).
8. Wein, Marcell, and Lestor Burtnyk, "Computer Generated Key-frame Animation", Journal SMPTE, (March 1971), pp. 149-153.