Computer Assisted Animation: 2D or not 2D?

J. W. PATTERSON AND P. J. WILLIS

John Patterson† and Philip Willis‡

†Department of Computing Science, University of Glasgow, Glasgow, G12 8QQ, UK

‡Computing Group, School of Mathematical Sciences, University of Bath, Bath BA2 7AY, UK

2D animation can only be automated to the extent that the computer acts as an interactive assistant to the animator. The key problem is that the 3D information which is implicit in the animated drawings is unavailable and this has encouraged the view of 2D animation being a subset of 3D animation in its fullest generality. However, introducing more than the absolute minimum of 3D information, essentially that contained in a hierarchy of drawing overlays, has matching advantages and disadvantages because animators, for aesthetic reasons, deliberately break the rules of geometry and physics as they apply to real-world objects. A software environment which only supports 2D functionality and drawing overlays is sufficient to promote cost-effectively the full range of effects that animators use while the state of the 3D animation art is as yet incapable of this. Essentially the same techniques can be used on live-capture images once they have had a structure similar to that required for animation imposed on them.

Received September 1994

1. INTRODUCTION

We are interested in the problems of using computers to create animation, especially the traditional cartoon style of animation associated with the likes of Disney Studios. The term 'computer animation' has come to mean almost anything involving a computer and a moving picture [7,8,17]. Computer animation, CA, is thus the core technology for such areas as multimedia, scientific visualisation and virtual reality. We therefore use the term 'computer assisted animation', CAA, for the specific tasks in which the computer is used to aid in the production of cartoons.

CAA is thus a subset of CA and is a core technology for multimedia only. In our case it is about creating and manipulating 2D images, which includes images captured from the real world and 2D morphing as well as cartoon production. The 3D case is closer to claymation than to drawn animation. We of course recognize that there are many people investigating 3D modelling and animation and related computer techniques but our concern here is with the 2D case.

Traditional drawn animation yields a two dimensional product but animators have a good understanding of the third dimension. This seems to have led some workers to view 2D as simply a limited version of 3D but in fact the truth is exactly the other way round. It is 3D that is the simpler case computationally. Animators may have a good feel for three-dimensional space but they do not mimic reality. Animation exaggerates reality, not only in terms of features but also in terms of timing. Animation is deliberately not consistent, for example in showing the relative position of eyes on a head, or even arms around a body as the character is seen from different positions.

Figure 1 shows (a) a cartoon mouse's head face-on (b) a geometrically accurate side view, and (c) the same view as an animator is likely to draw it. The orientations of the facial elements, and in particular the fact that both eyes can be seen in (c) but not (b) shows the deliberate inconsistencies between the two views. Here the anatomical components have been drawn to best convey the sense of the action, rather than to be consistent with an accurate model of the imagined character. In short, animation does not respect geometry, unlike typical computer graphics.

Computer-assisted systems have been poor at addressing these fundamental problems as was pointed out by Catmull [3]. This paper discussed the problems of computer-assisted animation in largely negative terms. In part this was due to technological constraints: lack of peformance discouraged interactive techniques. Line-tests were arguably the stage of the process then best-suited to computer graphics but in practice even this moved into the domain of video recording: it is much simpler to video drawings, correctly aligned, than it is to digitise and re-align.

We think that the problems raised in Catmull's paper now need re-addressing. As we hope to show, there are many reasons to be much more hopeful, not just because computer technology has moved on but also because some of the claims are now seen not to hold up. It is our firm belief that computer-assisted animation has arrived, offers solutions which are not all based in conventional computer graphics and can be made to deliver economically in a real animation studio.

2. THE ANIMATION DATA-PATH

We should start by giving an overview of how a drawn animation is prepared. What we give here is a relatively full account as it might be in a large studio. Some of the steps described might be simplified or largely omitted in a small studio which perhaps only works on single, short animations. We think it is important to understand the

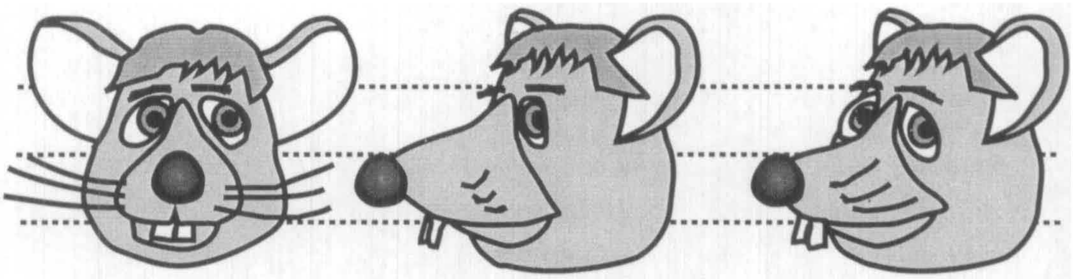

FIGURE 1. (a) a cartoon mouse's head face-on (b) a geometrically accurate side view, and (c) the same view as an animator is likely to draw it.

full process however, as that is what a commercial computer system must aim for.

The overall work can be thought of as falling into two phases, that of design (which might be about 15% of the total effort) and implementation (about 85%). We have tried to show the various logical components of the production process, but there is often overlap and even parallelism.

2.1. Story development

Whether from an original concept or by buying rights to a book, a story must be developed to fit the proposed duration of the movie. This in turn will be refined into a storyboard. A storyboard shows sketches of the envisaged movie, each of which represents perhaps 4-5 seconds of action, with the dialogue shown against each picture. A shooting script and a timing chart are also produced.

Model sheets are also prepared. These show the major characters in different poses, from various angles and with a variety of expressions. They will also show the colouring to be used. In effect, the model sheet is the nearest an animator usually has to a sculpture of the character. The model sheet is used to fix in the animator's mind how the character moves and what it looks like.

This completes the design phase.

2.2. Leica test

This somewhat dated term is used for the first attempt to put together a film of the required length. As there is no animation available yet, it is filmed from stills corresponding to the storyboard, but each still is held for as long as the corresponding sequence will take. A soundtrack will be added because the animators will work from the timing of the sound, especially for dialogue but also for overall pace. The result gives an impression of the movie. Weaknesses can be spotted and a re-design can take place to rectify them.

2.3. Scene staging

Scene staging is the mapping out of each scene, in particular the relationship between the characters and the background elements. The backgrounds may well be prepared in full once this has stabilised. Backgrounds are those elements, typically scenic, which are unvarying across many frames and so can be painted in much more detail than can a character. A good background can give a lot of visual richness to a scene or can set a mood, as well as providing the correct context and contrast for the main action. Backgrounds are often much larger than the frame size, to allow for pan or zoom effects.

The term 'staging' is usually reserved [6] for the development of the viewpoint the animator is encouraging the viewer to take. Staging is really the clear presentation of an idea which includes avoiding anything which might distract the viewer's attention or placing characters, viewpoints, expressions etc, even the timing of a scene, so that the viewer's attention is engaged most directly. This use of the term 'staging' really covers the design issues in setting a scene while scene staging is about how to implement the design and may involve questions of how to break down the scene into layers.

2.4. Exposure sheet

An exposure sheet is basically a table which has a row for each frame. The columns show which cels are used, the rostrum camera positions, instructions for exposure etc. Sometimes pairs of rows are identical ('shooting on twos') to show that successive frames in each pair are identical, this being adequate for most movement and a great saving in time and effort. The exposure sheet forms the complete instructions that the rostrum camera operator requires to assemble the final movie.

Every artwork column in the exposure sheet will also have an associated route sheet. This is a log of the production state of each artwork sequence. It records for each piece of artwork whether it is a key frame, a breakdown frame or an in-between, as well as whether the sequence has been line-tested, cleaned up etc.

To prepare this documentation, it is necessary to break down the scene into cel layers, including the separation of character components into layers. The number of layers cannot be too high because of the cels being neither perfectly transparent nor perfectly thin. Breakdown into layers allows the individual pieces of artwork to be named and indexed as they are produced. An exposure sheet track in which the artwork changes every frame, for example, will have a code to identify the overall sequence and another numerical code to identify which drawing within the sequence. This is always possible because the timings are fully known at this stage.

2.5. Drawing

The next stage is thus to produce the drawings for each cel in each frame. These are line drawings, not coloured, and prepared on something which resembles tracing paper with accurately-located holes for mechanical alignment.

This is done in three phases. Firstly the extreme drawings are produced. These show the major features of the action and are drawn by the main animator. Next, assistant animators, almost as skilled as the main animator, break down the action between the extremes, typically every four frames or so. Finally, the most junior animators produce the in-betweens to yield each frame. Extremes and breakdown drawings are usually quite rough, with lots of experimental over-drawing, and will usually have to go through a clean-up stage.

2.6. Line test

A line test is shot (nowadays often onto video) without the backgrounds, and still using the line artwork The purpose is to verify that the movements are correct and that characters interact accurately. In some cases, this may reveal problems and those sections will have to be reworked. If the line test is accepted, and on film rather than video, it can then be spliced into the Leica test, replacing the corresponding section of stills. The Leica test thus evolves towards the complete movie. Indeed those of us who are used to computers have to remind ourselves that the product at any stage is simply the current piece of film: the task of recreating this from the drawings is much greater than that of splicing together computer files. At the end of the process, the film is the totality of what is required: all of the drawings can be disposed of.

2.7. Ink and paint

The outlines are copied onto cels (transparent acetate sheets) and colouring is performed on the back of each picture. Xerography may be used to assist or replace the inking process. Colour consistency is a practical difficulty: all characters will have their colours precisely defined so that they can be mixed from standard paints. Special paints have to be used to draw on cels, in contrast to backgrounds where most materials can be employed. Translucent paints are available for special effects but otherwise high opacity is needed.

2.8. Rostrum camera

The artwork for each frame is now brought together on the rostrum camera, illuminated and shot, possibly incorporating photographic special effects. This film may also be spliced into the Leica test, replacing the corresponding line sequence, as it evolves towards the finished movie.

Rostrum cameras are physically massive and offer some facility for moving backgrounds separately and precisely. Rear and forward lighting may be available. The camera normally points vertically downwards and the artwork is placed horizontally. For certain special results, a multi-plane camera may be used. This normally operates horizontally and, as its name suggests, allows artwork to be placed in a number of well-separated planes to give a good feeling of depth. It is thus most often used for scene-setting shots, such as the opening, with the camera moving across of into the depth of the scene.

2.9. Soundtrack

The final soundtrack is then synchronised with, and added to, the movie. At the outset, the sound was used to drive the animation timing: in the end, the fine adjustments needed are made the other way round.

3. COMMERCIAL CONSIDERATIONS

It is rather easy to assume that because something can be done on a computer, so it follows that it should be. In fact the issue is not as clear cut as that, certainly where animation is concerned.

These days, the main market for animation is television and the budgets are usually small. Cost-effectiveness therefore dominates the thinking of the purchaser. Hand-drawn animation relies on a large number of people but only a small number are at the highest skill level. It is therefore the case that low-wage economies with capable artisans are extremely competitive sources of the mass-production aspects of animation, namely making and painting the in-between drawings. The industry has therefore seen a trend in which this work is sub-contracted to developing countries in the Pacific rim, while the main design and planning takes place in the West.

On purely commercial grounds, it follows that a computer system must compete not merely with an old-style animation studio with hundreds of direct employees but with the modern small studio which is sub-contracting heavily to low-price labour.

With the sub-contractor arrangement, the opportunity to enter the industry at the apprentice level also evaporates in high-wage economies, which in turn means that the skills base dries up until there are no senior animators either. In due course, this may lead to a situation where the entire process takes place in developing countries *even if* it is computer-based. The economic argument for using computers is thus not resolved. The cheaper computer systems become, the more this becomes true.

We therefore need to look beyond direct costs. As the preceeding section shows, computer animation is not mainly about drawing. There is a heavy element of pure organization required. A complete computer animation package must therefore address the managerial aspects too. The main benefits are therefore likely to include the uniformity of processing, cross-referencing the various

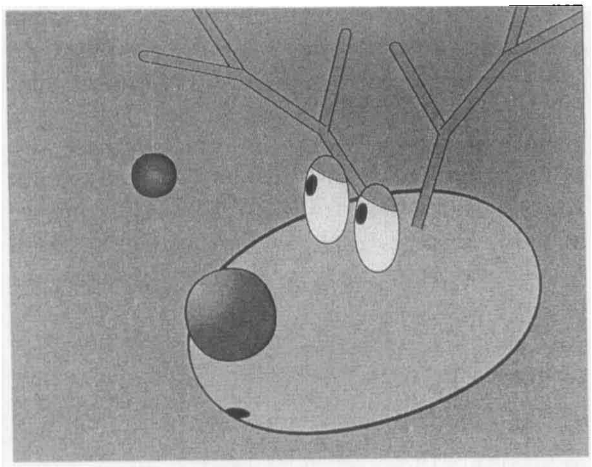

FIGURE 2. CAA rendering of Reindeer's head.

stages, database techniques for frames and sequences, the ability to capture and reuse movements, the separation of visual appearance from movement and sound, and sharing of load (both human and computational). Less obvious perhaps is the potential for standards to emerge, whether *de facto* or official, allowing the development of human skills in the new technology.

3.1. Manipulation of photography

Less obvious still is the way in which a computer implementation of a manual process might make possible tasks which would be out of the question to do by hand.

Some examples are shown in Figure 2. Here we see the use of graduated fills with arbitrary centres within the circular boundaries and a variable ink line around the border of the elliptical (head) region. Neither of these effects can be achieved to anything like that degree of control by hand.

If the in-betweening process is automatable then it should be possible to apply the same kinds of processes to images captured as, or from, photographs. In fact there are a number of implementations of so-called morphs [2] which use fairly crude in-betweening techniques to make one shape change into another. Since we have no real-world experience to compare the product of such a morphing process, we accept what we see as quasi-realistic no matter how unrealistic we would judge the in-betweening technique if used more conventionally. We also don't see the manual re-touching which is so often necessary to the raster image, for example re-touching backgrounds which get dragged around by the morphing process, in complete contradiction of what might happen in nature, as the silhouette changes.

The key point is that, although the photograph or *live capture* is a 2D representation of a 3D scene it is not necessary to recover that 3D information to be able to manipulate the image or use progressive transformations of it to make a movie sequence.

In fact image-processing filters can be used in the same way, a familiar example being the distortion of the region around the face of someone who wants to avoid recognition on TV, and a less-familar example is shown in the three images in Figure 3 a-c which show a 'teleport' sequence made entirely through using filters on an image then running the resulting sequence backwards.

4. INFLUENCE OF 3D

In fact, Catmull [3] argued that the main difficulty in automating CAA is that the 2D picture does not have the 3D information which the animator holds mentally. Drawings are just stylised 2D representations of 3D images and we expect them to behave the way our 3D mental models do, although animators regularly trick our visual system by breaking this link to achieve certain qualities in a movement. The key problem in 2D animation is how to automate in-betweening: the process of generating successive drawings of a figure which change consistently with our 3D intuition of how the drawings should change. Essentially this breaks down into two sub-problems: how silhouettes change (for example our hero runs into a wall and is flattened by the impact), and how the various parts of the figure

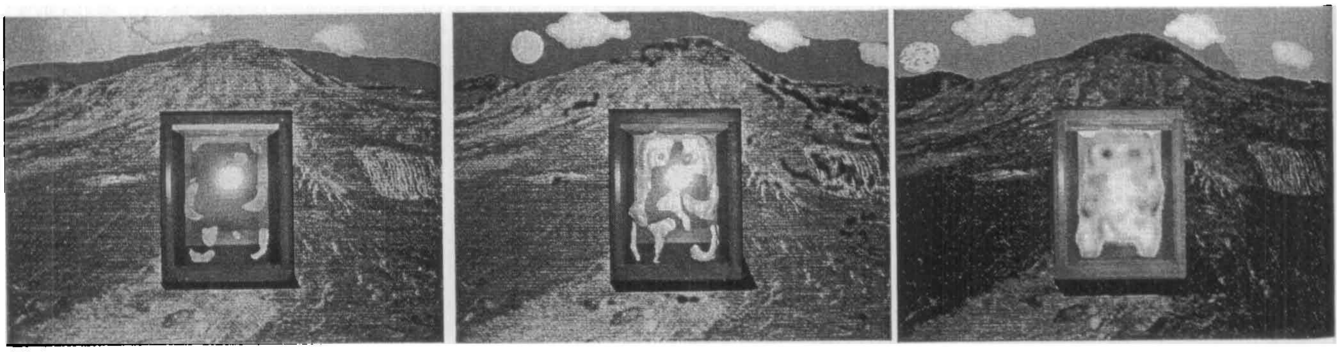

FIGURE 3. (a)-(c) Teleport Sequence from *Eric the Untenable*

occlude themselves (for example by head, body, or viewpoint rotations).

4.1. Skeleton in-betweening

The changing silhouette problem is the one addressed directly by so-called in-betweening systems. These are essentially of two types, skeleton-based or shape-distortion based. The appropriate real-world model for the skeleton-based approach is the kind of paper puppet made famous by 'Captain Pugwash'. The idea is to have a skeleton of lines which, in a human figure, would look like a stick-man drawing. The lines behave like bones (which can bend and stretch) and are normally jointed at the places one would expect. The silhouette is adjusted to follow the movement of the lines. Essentially these lines play a similar role to 'snakes' in image analysis [18].

The problem with pure skeleton-based methods is that they require a lot of effort to set up. The process of reconstructing the silhouette does not work well around joints and may require some manual reworking. Also some distortion model is required to make sure the silhouette changes with staging, for example when the skeleton is distorted or any part of the figure is rotated with respect to the viewpoint. Even so, one of us has experimented with skeleton-based animation [13] with some success. The principal benefits of this approach are the ability to store and mix character movements, as defined by the changing position of the skeleton; and the separation of appearance from movement, allowing independent re-use of these two elements. To present an adequately fluid visual appearance, the various shapes making the character are themselves interpolated and must therefore be represented in a very general way [12]. In short, we can use the skeleton for the gross position and orientation of the visual components but we need a second process to distort the overlying shapes as the character moves. Typical 3D systems do not address this second process, another example of 3D being a special case of 2D, rather than the other way round.

4.2. Shape-distortion in-betweening

Just as the skeleton method in the end requires the shape-distortions to work well, so a shape-distortion method relies on a hierarchy of drawing elements, or a Hierarchical Display Model, HDM, to provide a completely general in-betweening capability. When HDMs are used to structure and order the rendering of artwork they have gone under other names, for example Shape Tree, [13] and Hierarchical Structure Network, [5] but we will stick to the generic term HDM [16] [4] which captures all the specifics required here.

By way of an example we show in Figure 4 a possible HDM for the mouse head from Figure 1(a). The rule for implicit composition is to place the composition of the right subtree into the image first, followed by the composition of the left subtree, which has the opportunity to overlay and obscure parts of the right subtree

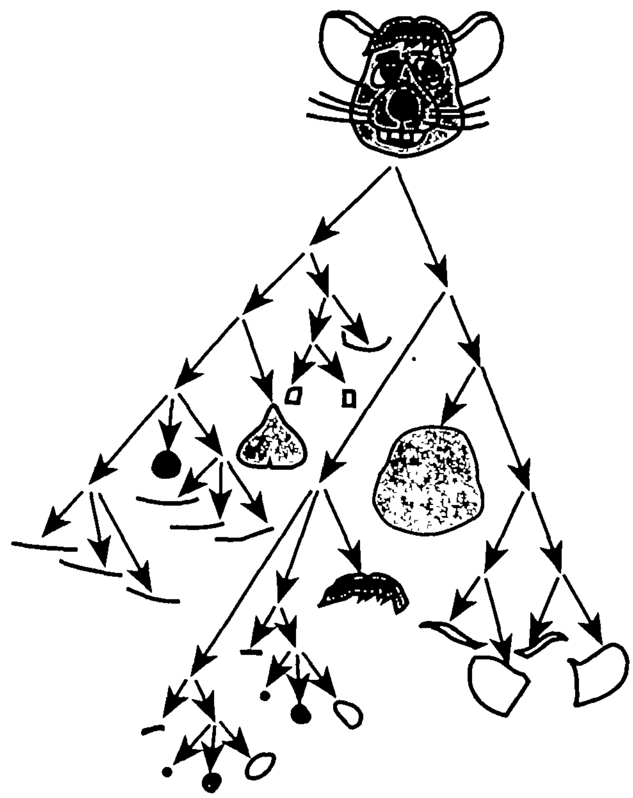

FIGURE 4. HDM for Mouse's Head.

composition. The hierarchy in Figure 4 reflects the grouping of facial elements more precisely than would be necessary for animation. For example the three shapes defining an eye are grouped together to form a subtree whereas the implicit 'over' composition rule in HDMs could be relied on to allow all the elements which are placed over the face silhouette to be shown at the same level, regardless of 'natural' grouping. A 'realistic' HDM would be much flatter than the one shown.

The only shape-distortion technique Catmull [3] distinguishes is the linear interpolation of Cartesian coordinates, or CCLI, which he rightly identifies as being unsuitable for general use. CCLI does not preserve shapes or proportions, which can be easily verified by

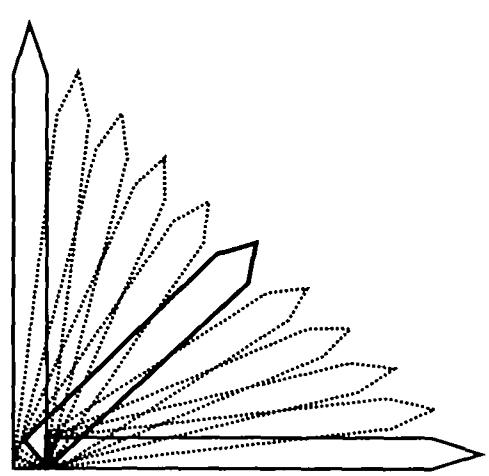

FIGURE 5. Pencil Falling in CCLI.

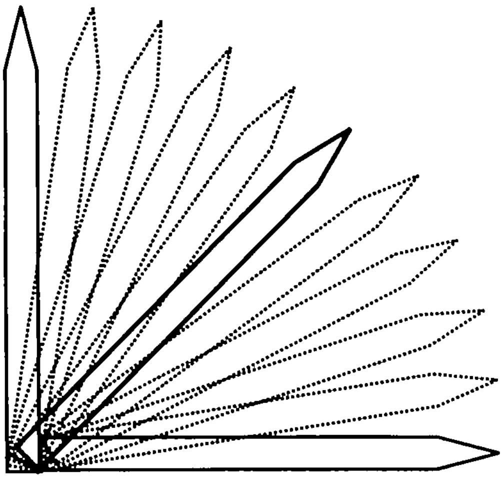

FIGURE 6. Pencil Falling in PCLI.

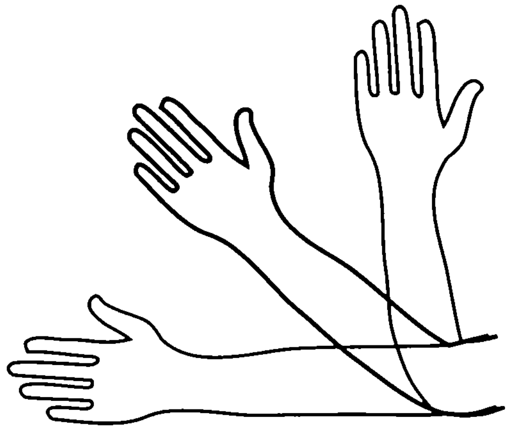

FIGURE 7. Hand moving about elbow (PCLI).

considering the in-betweening of a pencil falling from an upright to a reclining position, as in Figure 5.

The tip of the pencil will lie on a straight line between the two positions so no in-between will show of the pencil being the same length as in its key positions. Attempts to correct this, for example by editing the mid-point breakdown back to the correct length, simply results in new breakdowns with foreshortened proportions, which have to be edited again, etc. In reality the tip of the pencil describes a circle, as in Figure 6, and observation of the motions of cartoon figures shows that they nearly always make circular movements. This is partly because limbs rotate about joints and make circular motions and partly because animators are trained not to make straight-line movements (even for head-turning) because these are perceived to be less visually interesting than movements which have a circular sense to them [21].

More recently it has been suggested that [22] the linear interpolation of polar coordinates, or PCLI, is a more suitable shape-distortion technique because it naturally tends to respect shapes and produce circular motions, as does the hand in Figure 7.

Here the centre of the coordinate system is replaced relative to the previous point so this is really a relative polar coordinate scheme. The simplest way of obtaining these coordinates is to take the angle between the straight lines joining adjacent points defining the curve, and their lengths, all of which can be obtained straightforwardly from a vector model, so both lengths and angles are taken relative to the previous point in any traversal of a curve represented by these points. This is referred to in CAD technology as the *intrinsic* method of curve definition [1]. However by itself this technique is insufficient to deal with the more difficult self-occlusion problems, for example silhouette changes during head-turning.

4.3. Hierarchies

Self-occlusion is the most serious obstacle to any purely 2D, or drawing-based, automatic in-betweening strategy. In many cases it can be dealt with simply by overlaying elements of the drawing and progressively covering or uncovering the elements as part of the in- betweening process. These elements form the leaves of an HDM and the order of placement of the elements (or cutting operations involving them) in effect re-introduces 3D information which is not readily apparent in the composite drawing. Here the HDM is acting as a model for the figure although it is only valid for a limited range of angles through which the figure might be perceived to be turning. A collection of HDMs for the same character forms the main entries for an electronic model sheet for that character and, in effect, substitutes for a true 3D model from which HDMs could otherwise be deduced. Using model sheets like this is usually referred to as 2½D modelling—not 2D but not really 3D either—although all an animator needs to do is to select an HDM from a well-enough populated model sheet to start building a sequence. There is an assumption here that all the HDMs needed for a figure need to be put in by hand but there are techniques from image analysis which could help in building and verifying HDMs automatically.

4.4. Continuity

Although it would be possible to solve all self-occlusion problems using a suitably well-defined HDM some problems, particularly those involving discontinuities in shape-distortion, are not conveniently handled that way. Setting up in-betweening problems, or steering them to a resolution, is a critically important part of realising the economic advantages of CAA, so it is essential to uncover, and make use of, convenient techniques wherever possible. Here it is much more convenient to deal with the discontinuity directly. Let us assume for the moment that we are in-betweening drawings made from matching HDMs and that the elements of the drawings are made up out of polylines. (These polylines may be decorated with attributes which determine the local appearance of the line.) A first in-between—the 'mid-point'—can be produced using the PCLI model by interpolating the lengths and angles of the polyline. If this mid-point does not strike the pose which the animator would expect then it can be edited by adjusting these lengths and angles. Now the in-between does not correspond to any single value for the interpolant although it is possible to calculate straightforwardly what the interpolant should have been for each of the interpolated parameters. It is now possible to produce a motion which passes smoothly through the edited drawing by treating each interpolant individually and interpolating the interpolants themselves with, say, a spline [19]. This gives us a smooth motion which will be C^2 continuous in time if we have at least four interpolant values through which to run a C^2 continuous spline. However, we do not want all in-betweens to develop from one another with C^2 continuity in time, but if our spline model allowed the specification of lower orders of continuity, for example C^0, then we could have our shape changing its silhouette smoothly, then experiencing a C^0 continuous event which led to a discontinuity in the shape distortion process. Such discontinuities are typical of the self-occlusion of irregularly-shaped objects.

4.5. Path direction and continuity

The type of spline we need is a non-uniform B-spline with non-replicated knot-vector entries being determined by as close an approximation to arc-length parameterisation as is practicable. Similarly the drawing elements themselves need to be defined by non-uniform splines and it is the control point polygon which replaces the polyline in our foregoing description. Again we need to have full local control of continuity during edits. This model supports a process in which in-betweens are produced individually and edited if necessary. Edits may involve altering line trajectories or altering the HDM itself, and any degree of continuity of movement can be maintained through an edit. Since the linearity of the interpolation has now long gone we refer to this method as just 'polar interpolation', or PI.

An extreme example of this involves the case in which one might want to traverse a curve in the opposite direction after a given position has been reached. At the redefinition position the curve may be ambiguously traversed in either direction. An example is shown in the situation of the mouse's head turning from left to right, as in Figures 8–10.

Here three keys are shown, the left-facing extreme (Figure 8), the face-on position (Figure 9), and the right-facing extreme (Figure 10). (We may note in passing that the extremes are geometrically inconsistent with the face-on view and that the nose makes a circular, rather than a linear, trajectory from left to right. None of these contrivances has any impact on the resolution of the problem.) In what follows we will not distinguish manipulating the control point polygon from manipulating the curve. Traversing the curve involves encountering the points in a certain order, so traversing the curve in the opposite direction involves encountering the points in a different order, or sequencing the point indices differently. Figure 8(a) shows the traversal directions for the mouse's muzzle parts, which are reversed in 10(a), while the traversal of the head is always in the same direction. The cross-over point, when both traversal directions can be applied to the same face-part is the face-on position, 9(a). Since PI is a relative reconstruction technique this means that the interpolants associated with a given index have to experience a C^{-1} discontinuity to ensure they match the values they are replacing. However, tangents (C^1) and rate of change of tangents (C^2) can be carried across from the matching interpolants so any degree of continuity can be preserved. In practice this formal complication can be avoided by tying the interpolant to the parameter being interpolated rather than its index in a traversal of the curve affected by that parameter. Other curve appearance parameters would have to be treated similarly so this is not a special case.

4.6. Using an in-betweener

The other assumption we made about matching HDMs is sustained if we work, in effect, from the same HDM. There are two models for in-betweening, *pose-to-pose* and *step-ahead* [20]. In pose-to-pose the action moves from an initial pose to a final pose (known as *keyframes*). If one pose is created out of the other by editing, the HDMs will largely correspond as there is a natural tendency on the part of the artist to retain as much commonality as possible for convenience and speed of working. There may be scope for building matching HDMs from independently derived drawings (the 'correspondence problem') but no good methods for doing this are known to us. Step-ahead in-betweening starts from a single pose and involves making repeated

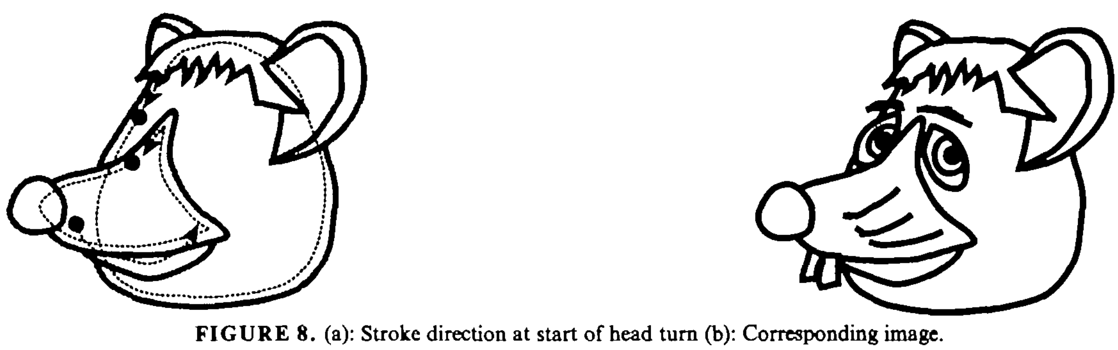

FIGURE 8. (a): Stroke direction at start of head turn (b): Corresponding image.

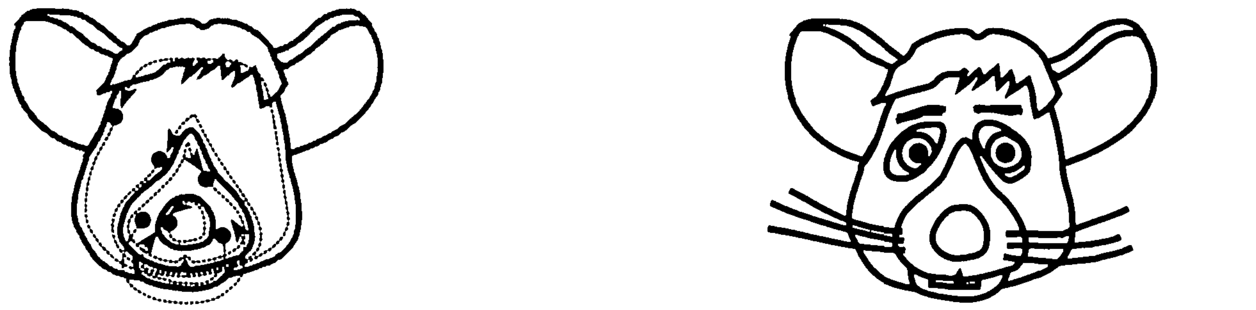

FIGURE 9. (a): Ambiguous stroke direction (b): Corresponding image.

edits followed by continuity-guided extrapolations. The pose-to-pose methods we have described here can accommodate this style of in-betweening quite readily as it is straightforward to drive interpolant interpolation to provide interpolant extrapolations.

There is one particular case where hierarchies are especially valuable, namely in simulating the effects of a multiplane camera. Here the sense of the third dimension, which this technique so powerfully suggests, can be reinforced by simulating depth of focus in rendering the scene. This is taken directly from 3D graphics technology.

4.7. Paint-and-compose

The rostrum camera stage is one which is commonly modelled in commercial systems, for example Macro-Media's 'Director'. Director is designed around an exposure sheet and, although it offers capabilities which are beyond any mechanical process, essentially automates that stage of the animation data-path. Drawings have to be inked and painted individually and no drawing in-betweening is provided. All artwork is treated as uninterpreted rasters, save for a pixel coverage measure which could possibly be used for transparency as well. Such a system is known as a 'paint-and-compose' system, a term which also covers systems which might try to automate inking and painting by trying to match parts of adjacent drawings and so deduce how to colour succeeding drawings. This is not a reliable process. Where in-betweening is going to be employed, artwork must be held in an HDM format. This is often referred to as an object-oriented format because the HDM mimics the object structure and hierarchy. As we have seen HDMs can be in-betweened and also decorated with inking and painting attributes so that automatic inking and painting can be done reliably. In fact the economic justification of providing in-betweening is largely in terms of its contribution to reducing costs further along the animation data-path, which is so large that in-betweening interaction costs come to dominate CAA costs wherever it is provided.

We have developed a rendering system which is geared towards layering of cels, each of which is held in an HDM format [10]. This provides a number of advantages over traditional paint-and-compose systems. For example the use of an HDM readily permits computer-based operations to be performed, including both affine and non-affine transforms. In addition we use a general model of paint, allowing translucency to be modelled well [11]. Finally, we incorporate a more sophisticated lighting model than other systems, including front and back lighting of each cel separately. It is this 'best of both worlds' approach which we firmly believe will yield the maximum benefit from bringing computing and traditional animation practice together.

4.8. Animation databases

A paint-and-compose system is typically designed to support the building of single sequences and an exposure sheet is thus a suitable model for organising the artwork database. When in-betweening is involved, model sheets are also needed in the database and these can be called on, or contributed to, throughout the making of the movie, or even a series. So the database for an inbetweener-based CAA system needs to carry artwork for the entire project, which is nowadays within the capabilities of file-stores and database systems. Here the Leica test is a natural backbone around which to

FIGURE 10. (a): Stroke direction at end of head turn (b): Corresponding image.

organise the database. Although Leica tests have been made more-or-less obsolete by the use of videotape in studios, the idea is getting a new lease of life with the advent of CAA technology.

5. RE-INTRODUCTION OF 3D INFORMATION

So, the key elements to an effective in-betweening strategy involve PI on non-uniform B-splines, control of continuity, and manipulation of HDMs. These HDMs are in effect a half-way house between 2D and 3D. What if, instead of the $2\frac{1}{2}$D solution offered by model sheets, a full 3D model were used instead? If that were done then all the 3D information would be available and there would be no need for lots of poses with their associated HDMs, nor for 2D shape distortions, nor any of the mechanisms invoked to fake in the missing 3D information. The first problem is that building 3D models is an expensive process although this could be mitigated by a kit which allowed different body types and different body plans to be built up comparatively easily.

5.1. Kinematics

The second problem is that posing the model is difficult. The animator is being asked to interact with a 2D representation of a 3D model with tools which are only naturally meaningful in the 2D environment. It is a common experience for posed figures to float up from the floor if placed there at the beginning of a sequence. The usual solution to this problem is to model the figure's kinematics. The foot-on-floor problem can be dealt with by inverse kinematics—the solution starts with the requirement that the foot be on the floor and works backwards to the significance this has for each joint upwards through the figure. Kinematics-only systems are concerned solely with satisfying geometric constraints and tend to provide a wide range of solutions of which few are convincing. Nevertheless kinematics-based systems are available and are quite usable although they do hot produce solutions which are good enough for professional character animators.

5.2. Dynamics

More sophisticated still is to use dynamics rather than kinematics as the model for constraining the poses of the figure throughout the movement. Here the masses and other physical characteristics the 3D model would have in reality are simulated, so objects start to slide down slopes when gravity overcomes friction, and bounce off one another under the influence of impulses generated at the moment of collision. Although this approach has the advantage that the solutions it offers are much more like what one would expect in reality, it has the disadvantage that quite a lot of physics has to be simulated with the attendant simulation costs. 'Lightweight' dynamics simulations which minimise the dynamics which has to be taken into account, or which try to approximate it to cut down on computation, are possible and come close enough to simulate reality to make fully general calculations unnecessary for the accuracy required for animation.

5.3. 3D consistency

Unfortunately for these methods it is quite common for animators not to respect the very properties these systems try to simulate accurately. Kinematics is about geometry, which animators often don't respect, and the dynamics which supposedly apply to a figure may change within a single sequence. How does one deal with a sequence in which a character reaches for an object and, in the frame before he grabs the object, his hand is suddenly enlarged to the size of his body? This is typical of the sorts of trick that animators use to make an action seem more 'snappy'. The eye doesn't even register the hand size change properly but is left with the impression of a snatching action. By contrast, dynamics simulations slow down the action to speeds we are familiar with in live action, but which seem dull indeed in the accelerated world of cartoon animation. (This was also a criticism of early 2D animation systems which used CCLI, but for different reasons. The retained linearities in the movements were perceived not so much as being linear but as being 'uninteresting'—the very reason why character animators prefer to use circular motions.) Finally, the ways in which faces are animated are a major ingredient in viewers' perception of the quality of animation, yet this is still very much a research topic at the time of writing.

Animating faces is a specialised art in itself. Much of how we react to real humans is driven by the facial expressions and emotions that we perceive in others. For this reason facial animation has attracted significant research interest in the 3D community, for example Patel's work [14] [15]. There is a cautious need for some of this expertise to spill over into the 2D world. However we again point out that the 2D animator is not really mimicing human emotions but rather inventing a highly stylised set. The 2D character's eye-brows can shoot above the head, if sufficiently startled, while the facial droop of a down-hearted character has no counterpart in humans. A cartoon character shouting loudly may show its dentition literally outside the visage. For these and related reasons we believe that the underlying technology of 3D systems is more relevant than the sophistication of the facial control that they undoubtedly show. There is room for joint-working, certainly, but not for replacing 2D animation with a basically 3D system.

6. CONCLUSIONS

In this paper we have shown that, while it is essential to retain some information about the 3D structure of a cartoon figure, animators are disinclined to respect a faithful and consistent interpretation of its 3D model. Although the exploitation of geometry and physics can help to buttress the extremely high skill levels normally required of a top-quality animator this cannot be relied on routinely, so a base level of wholly 2D, or $2\frac{1}{2}$D functionality is always needed to fall back on. Since that functionality is unable to support any 3D property beyond occlusion it is necessary to appeal to the animator to correct the inevitable errors interactively. The key to cost-effective CAA, therefore, is to keep these appeals to a low enough level to be able to generate finished quality animation at a high daily rate.

6.1. Productivity

By hand an animator, with inexpensive equipment, can manage perhaps 1–3 seconds a day of finished animation, so a CAA system has to be able to match that at any wage level and pay for the computer system as well. A good CAA system should be able to manage 10–30 seconds per station per day.

The paint-and-compose systems we referred to earlier are normally only able to offer modest improvements in productivity. Where no interframe correspondence is used, the rate per animator is typically in the range 2–6 seconds a day. With interframe correspondence this gain is variable but could be in the range 3–10 seconds a day per animator. With an in-betweener, auto-ink and auto-paint become reliable and there may be some gain from automating the in-betweener, for which we estimate a range of 7–15 seconds a day per animator. The higher rates we quoted in the previous paragraph assumed that, in addition to automating in-betweening we also automate the process of analysing key drawings. This is a problem which also breaks down into three parts, vectorisation (finding the intended curve trajectory from the animator's rough outline [9]), path matching (finding the degree of continuity between line segment pairs at joints), and line labelling (determining the ordering of line strokes to build the HDM). The problem is compounded by the roughness of the drawings. It is possible, and essential for the economics of line analysis, to avoid having to go through a manual clean-up process, but this is at the cost of doing some re-working of the analysis. Such reworking is usually needed in any case because the path matching and line ordering processes are themselves not wholly reliable. Nevertheless we would expect a factor of two reduction in time at this stage, including any reworking time, which gives us the higher figures we quoted earlier.

6.2. Animating live action

Animation techniques are also used to create special effects in live action. It is far from unusual to use skilfully-painted images (on glass with cut-outs for inserting the action) as backgrounds and similarly animation may be overlaid on live action during the composition stage. Here the problem is that the backgrounds are usually treated as rasters and these can be enormous, perhaps as large as 96–128 Mbytes uncompressed, if they are going to be used as film backgrounds where even slight loss of resolution is exaggerated on the big screen. These sizes are quite manageable on modern fast workstations with multi-gigabyte filestores, but up till quite recently this kind of storage—and processing—demand was regarded as impracticably large. Other examples include so-called 'morphs', which are usually implemented as conformal mappings between sampled spaces combined with a colour blend, but the intermediate dimensions are usually determined by CCLI. It is now quite possible to contemplate performing image analysis steps to tesselate the image, with some of the tesselations groups hopefully matching recognisable objects, and structure the raster, which is just a series of colour samples up to this point, so that it can participate in the same operations that can be allied to synthetic animation. So CAA technology is actually a technology which can be applied in its entirety to any visual medium, that is, it is a multimedia technology and not restricted to cartoon animation. Again there is an interplay between 2D and 3D. While it is possible to treat the analysis of live action entirely in terms of an imposed structure from image analysis, it is also possible to recognise shapes and features in the image if there are models or prototypes of the shapes in the analyser already. It is still a research issue as to just how accurate—or generic—the models have to be to be able to perform good enough analysis.

6.3. 2D and 3D

After sixteen years Catmull's paper can be seen to have echoed the frustrations imposed on him by the technologies of the day. However, his description of the problems have been taken by many since to constitute timeless objections which were unlikely to be overcome by straightforward improvements in technology. To-day the situation seems different. Catmull's later conclusion was that the future lay in 3D animation because that was something which could only be done by computer, and this has resulted in the sophisticated 3D animation systems which are commonplace to-day. Only now are we beginning to see production-quality 2D animation systems as commercial products. Because of their relative neglect there are substantial research issues still to be addressed and the fact that these issues have not been addressed yet will impact the development of the burgeoning multimedia industry.

We started this paper with a deliberately punny title. It is our belief that 2D animation systems will become as ubiquitous as their 3D counterparts and that the distinction between them will become increasingly blurred as they use progressively more common techniques. However, by maintaining a conceptual distance from 3D systems, 2D CAA has developed a functionality which can ultimately be used to manipulate live action as freely as synthetic elements. For now CAA is 2D, and is, indeed, about to come into its own.

REFERENCES

[1] J A Adams. The Intrinsic Method for Curve Definition. *Computer Aided Design*, 7(4): 243–249, 1975
[2] T Beier. Feature-Based Image Metamorphosis. Computer Graphics (SIGGRAPH'92) 26 (2), pp 35–42
[3] E Catmull. The Problems of Computer-Assisted Animation. Computer Graphics (SIGGRAPH'78) 12 (3) August 1978, pp 348–353
[4] G Cockton. Architecture and Abstraction in User Interface Management. PhD Thesis, Dept of Computer Science, Heriot-Watt Unversity, 1993
[5] J. Foley et al. Computer Graphics: Principles and Practice. Addison Wesley (Reading Mass. 1990)
[6] J Lassiter. Principles of Traditional Animation Applied to 3D Computer Animation. Computer Graphics (SIGGRAPH'87) 21, 4 July 1987, pp 35–44
[7] N Magnenat-Thalmann and D Thalmann. Computer Animation: Theory and Practice. Springer-Verlag, Berlin 1985
[8] N Magnenat-Thalmann and D Thalmann. An Indexed Bibliography on Computer Animation. IEEE CG&A, 1985
[9] D Marr and E Hildreth. Theory of Edge Detection In R Kasturi and R Jain, eds, *Computer Vision* IEEE, Los Alamitos, 1991
[10] T Nettleship and P J Willis. The Animachine Renderer University of Bath, Computing Group Progress Report, Dec 1993.
[11] R J Oddy and P J Willis. A Physically-based Colour Model Eurographics UK Conference Proceedings, April 10-12th 1991, pp. 87–103. Appeared in: Computer Graphics Forum 10 (2) 1991, pp. 121–127.
[12] R J Oddy and P J Willis. Rendering NURB Regions for 2D Animation. Computer Graphics Forum, 11 (3), pp. C-35 – C44 and p. C-465. (Proceedings issue: Eurographics 92 Conference, Cambridge, Sept 1992)
[13] M Owen and P Willis. Modelling and Interpolating Cartoon Characters Proceedings of Computer Animation '94, Geneva, IEEE May 25-28th, 1994. Text pp. 148–155; colour plates p. 203.
[14] M Patel and P J Willis. FACES: The Facial Animation, Construction and Editing System. Proceedings of Eurographics 91 Conference, Sept 1991, pp 33–45.
[15] M Patel. Colouration Issues in Computer Generated Facial Animation. To appear in Computer Graphics Forum
[16] J Patterson and G Cockton. Composing Hierarchically Structured Images. Computer Graphics Forum, 11 (3), pp. (Proceedings issue: Eurographics 92 Conference, Cambridge, Sept 1992)
[17] X Pueyo and D Tost. Survey of Computer Animation. Computer Graphics Forum 7 (4) December 1988, pp 281–300
[18] M Sonka et. al. Image Processing, Analysis and Machine Vision. Chapman & Hall, London, 1993
[19] S Steketee and N Badler. Parametric Keyframe Interpolation Incorporating Kinetic Adjustment and Phrasing Control. Computer Graphics (SIGGRAPH'85), 19 (3) July 1985, pp 255–262
[20] F Thomas and O Johnson. Disney Animation—The Illusion of Life. Abbeville Press (New York 1981)
[21] A White. The Animator's Workbook. Phaidon, Oxford, 1986
[22] Yu J-H. Inbetweening for Computer Animation Using Polar Coordinates Linear Interpolation Research Report 90/R23, (Sept 1990), Dept of Computing Science, University of Glasgow