



**UNIVERSITY**  
*of*  
**GLASGOW**

Yu, J-H. and Patterson, J.W. (1997) Assessment criteria for 2D shape transformations in animation. In, *Computer Animation '97 , 5-6 June 1997*, pages pp. 103-112, Geneva.

<http://eprints.gla.ac.uk/3543/>

# Assessment Criteria for 2D Shape Transformations in Animation

Jinhui Yu & John W. Patterson

Glasgow Interactive Systems cenTre, Dept of Computing Science  
University of GLASGOW, G12 8QQ, U.K.

## Abstract

*The assessment of 2D shape transformations (or morphing) for animation is a difficult task because it is a multi-dimensional problem. Existing morphing techniques pay most attention to shape information, interactive control and mathematical simplicity. This paper shows that it is not enough to use shape information only and we should consider other factors such as structure, dynamics, timing etc. The paper also shows that an overall objective assessment of morphing is impossible because factors such as timing are related to subjective judgement, yet local objective assessment criteria, e.g. based on shape, are available. We propose using "area preservation" as the shape criterion for the 2D case as an acceptable approximation to "volume preservation" in reality, and use it to establish cases in which a number of existing techniques give clearly incorrect results. The possibility of deriving objective assessment criteria for dynamics simulations and timing under certain conditions is discussed in the paper.*

## 1 Introduction

Many 2D shape transformation techniques for computer animation have been proposed in the past two decades. These tend to replicate the in-betweening process of hand drawn animation and work in the following way: when given two key shapes, they transform the shape continuously from one to the other. There are essentially two types of in-betweening systems: shape-based and skeleton-based. With the shape-based approach there are two categories: direct and indirect. In the first category (direct) the shape is transformed directly, as in Simple linear interpolation [1], which we refer to here as CCLI (Cartesian Coordinate Linear Interpolation); Inbetweening using moving point constraints [10]; Polar coordinates based linear interpolation techniques [15] [11], which we refer to here as PCLI; A physically based approach to 2D shape blending [12]. In the second category (in-

direct) the shape is transformed into some representation first, then interpolation is performed on certain parameters in the representation and finally the shape is reconstructed. Examples of such techniques are Polygon morphing using a multiresolution representation [3], Matching and interpolation of shapes using unions of circles [9], Implicit curved polygons [7], Solid-interpolating deformations [5]. In the skeleton approach, we first need to have a skeleton, which can be either devised manually or derived automatically from the shape to reflect the structural information of the object, then the skeleton is transformed, and finally a "skin" is put on to the skeleton in some models. In the 2D case only a few skeleton techniques have been proposed: Interactive skeleton techniques [2] and the star-skeleton representation [13]. Here the interpolations applied to the skeleton are CCLI and PCLI respectively. All the foregoing methods can produce correct intermediate shapes for some examples but fail to do so in others.

At present judgement of the correctness of an animation sequence generated by a morphing technique is entirely subjective: play it back and judge it in our minds. If the result looks wrong, we need to modify the transformation according to the information revealed in the playback phase, then generate the sequence and check it again. From this process we see that correct morphing is a vital step in automatic in-betweening.

Most objects we might meet in animation are living objects, natural objects, man-made objects, effects etc. Despite their different forms they all have structures. When they move, their dynamics are constrained by these structures. In animation, reality is exaggerated, not only in terms of features, but also in terms of timing. Thus for the correct morphing of a moving object, we should pay attention not only to *shape*, but also to *structure*, *dynamics* and *timing* and such other factors as required by the principles used in traditional animation.

Existing morphing algorithms pay much attention to shape information, interactive control and mathe-

mathematical simplicity. Of these only a few take structure information (the skeleton) into account. Since traditional animation does not pay a lot of attention to realistic dynamics and timing, correct results can only be achieved when the key shapes represent the structure, dynamics and timing of the movement at the same time.

Since morphing is a multi-dimensional problem, the assessment of a morphing technique should be made in multiple dimensions. This paper introduces criteria such as shape and dynamics for local objective assessments. We propose an “area preservation” principle as an acceptable approximation to “volume preservation” as our shape criterion, and the violation of this criterion seems to happen at the same places as certain existing morphing techniques are subjectively judged to have failed. A similar approach offers the possibility of objective assessment of dynamics and timing under certain conditions.

In this paper, we first describe how to express our shape criterion in algebraic form, then we apply the formalism to examples of the use of individual techniques and thus establish the conditions where intuition and algebra agree. Next, we discuss other factors such as dynamics, timing, movement continuity, path direction and continuity. Finally, we tie in effects animation which is usually thought of as being unrelated to existing morphing techniques and should be dealt with using a different approach.

## 2 Area preservation

In 2D computer animation, the object is represented by its silhouette and may be regarded as being approximated by a polyline or a polygon. (If, for example a B-spline curve is used it is rendered as line segments which approximate the curve to vanishing accuracy, so in the end everything is represented as polygons or polylines, however small.) In traditional animation, usually the object shape deforms but its features retain their character during the movement. A typical technique used in traditional animation is “squash and stretch” in which the object is stretched out in the air and squashed when it hits the ground. Commonly this technique is used to prevent strobing which will occur if silhouettes fail to overlap on successive frames, also the shape changes approximate the footprints of motion blur. In squash and stretch animators are taught to think in terms of preserving the volume of the shape, i.e. no matter how squashed or stretched out a particular object gets, its volume remains constant. This satisfies everybody’s intuition

as to how the world works, that is both animators and audience agree as to the correctness of what they are seeing, even though the technique is an exaggeration of reality. The reason for this is undoubtedly because the technique also matches the more esoteric point made earlier about blur footprints. We get away with squash and stretch because of this coincidence. On the other hand if we ignored volume preservation we risk not just strobing but shape misperception effects as well. For example, if an object is squashed down without its side stretching, it would appear to shrink, if it is stretched up without its sides squeezing in, it would appear to grow. This principle is taught to animators in terms of “volume” and in the 2D case experience has shown that this can be approximated by “area”, whence “area preservation”, and from this we can form our shape criterion to judge morphing results.

It is well known that, given the vertices of a polygon, the polygon’s area is determined by

$$S = \left( \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & y_2 \\ x_3 & y_3 \end{vmatrix} + \dots + \begin{vmatrix} x_{N-1} & y_{N-1} \\ x_N & y_N \end{vmatrix} + \begin{vmatrix} x_N & y_N \\ x_1 & y_1 \end{vmatrix} \right) / 2 \quad (1)$$

where  $S$  refers to the polygon area,  $x_i, y_i$  ( $i = 1, \dots, N$ ) are the coordinates of the  $i$ th vertex of the polygon, and  $N$  is the number of vertices contained in the polygon.

If we are given two key polygons  $P_1$  and  $P_2$ , we can calculate their corresponding areas  $S_1$  and  $S_2$ .  $S_1$  and  $S_2$  can either be the same or different, for example when zooming into an object. In most practical character and object animation the intermediate area should change progressively according to the foregoing principle. Geometrically this can be described in terms of an intermediate area  $S_t$  ( $0 < t < 1$ ) varying linearly from  $S_1$  to  $S_2$  as  $t$  goes from 0 to 1. Thus the criterion is that if  $S_t$  meets this condition then the result is correct, otherwise not.

The foregoing criterion is an ideal one, and can be described using a straight line drawn from  $S_1$  to  $S_2$  as shown in Figure 1. In practice, however, we should recall the “area preservation” is only an approximation we can get away with. We do not require that the area should be strictly preserved in the object deformed, so it is sensible for us to allow  $S_t$  to vary in a tolerant range along the ideal straight line. If  $S_t$  falls in the range we can still get an acceptable result. To obtain this tolerant range we first calculate the error between achieved intermediate areas and the ideal one as expressed by the following formula:

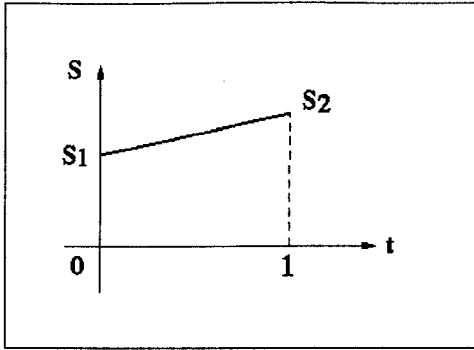


Figure 1: Area preservation criterion

$$Error(t) = (S_{at} - S_t)/S_t \quad (2)$$

where  $S_{at}$  refers to the achieved intermediate area and  $S_t$  refers to the ideal area which is derived by the linear interpolation between  $S_1$  and  $S_2$ . Next we explore the numerical conditions for the tolerant range. We take the area of an object in a correct animation sequence as a reference, then we distort the object thus causing a change in its area, and then play this back to see if it is acceptable. The test was done with different values of  $Error(t)$  and we found that if  $|Error(t)|$  was less than 10% then the result was acceptable. From this we have a numerical condition to apply.

For an object which is represented by a single curve (polyline) rather than a polygon, we calculate the length of the curve instead and thus the criterion turns to be “length preservation”. Here area parameters in equation (2) are replaced by length parameters, and we use the same acceptance criteria.

Usually area is preserved in skeleton techniques and, as long as the skeleton is correctly transformed, we can apply our “length preservation” criterion to the skeleton as before.

## 2.1 Local validity

When certain interpolation algorithms are applied to a polygon representing a structural object like a human figure, the outcomes include varying local distortions in intermediate shapes as shown in Figure 2. As a result, only example (c) with the star-skeleton technique looks right throughout the sequence. In other examples, the lower leg becomes shorter in (a) with CCLI, thinner in (b) with PCLI, or thicker in (d) with the multiresolution method, none of which are acceptable in character animation. Those examples actually pass our numerical condition because the resulting distortions are local and the error in the entire intermediate

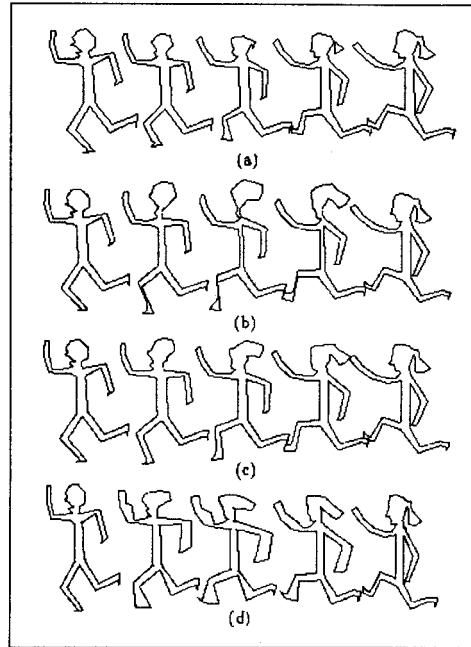


Figure 2: Comparative results of morphing algorithms applied to a run

area is less than 10%. Therefore our numeric condition fails in such cases.

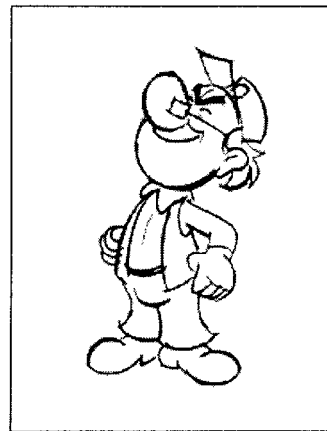


Figure 3: A practical cartoon character

In practice, however, this problem is not serious. Figure 3 [4] shows a typical human figure used in traditional animation. Here we can see that different (local) parts of the body are drawn with individual polygons or polylines and it is rare for an animator to draw the whole body with a single polygon as shown in Figure 2. Therefore we can apply our criterion locally to each individual polygon which tiles the figure and make separate assessments.

The foregoing criterion can be used as a general one to accept or reject the intermediate shapes generated by different morphing techniques. We know that each individual algorithm has an advantage or disadvantage in one aspect or another, for example CCLI is the simplest and the quickest in execution, but causes distortion when modelling rotation. Using our criterion we can derive applicability conditions for CCLI.

## 2.2 Where CCLI fails

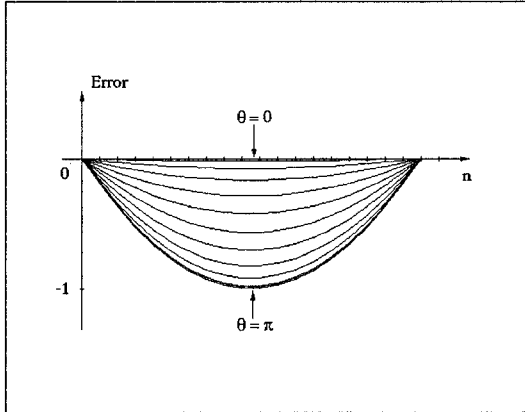


Figure 4: Area distortion error of CCLI under different rotation angles

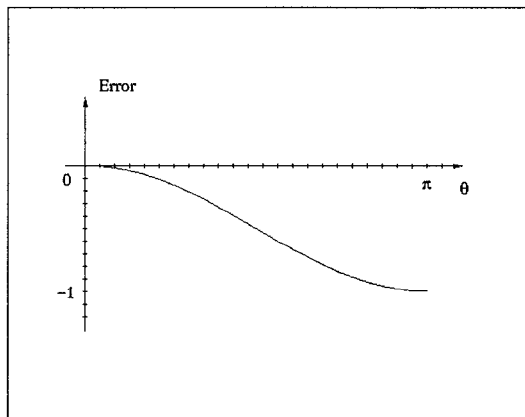


Figure 5: Area distortion error of CCLI against rotation

Figure 4 shows the area error of CCLI under different rotation angles between two key squares. When the rotation angle is 0 CCLI gives a perfect result with no distortion at all. As the rotation angle increases, the area error also increases and when the rotation angle is  $\pi$  radian the error reaches its maximum value -1. Here the intermediate polygon area is 0 which corresponds

to the worst distortion CCLI may cause in this example. Figure 5 shows the graph of error against rotation angle between two key shapes. Using our assessment criterion we see that 10% error corresponds to  $\pi/6$  radians in the horizontal axis, which means CCLI is applicable when the rotation angle is less than  $\pi/6$  radians between two key shapes.

We can also apply this condition to the skeleton in skeleton-based techniques.

How to calculate the rotation angle between two key shapes will be described later. In the case of the rotation angle exceeding  $\pi/6$  radian, a possible candidate interpolant is PCLI.

## 2.3 Where PCLI fails

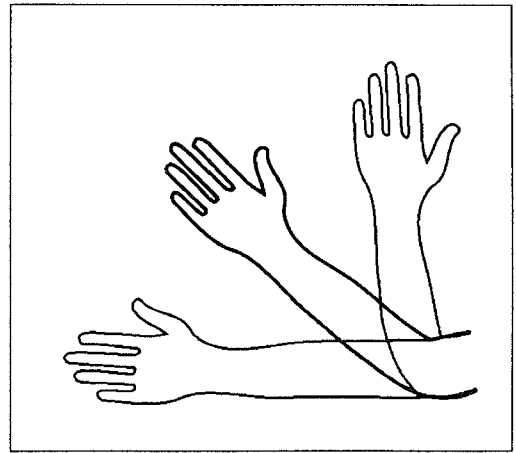


Figure 6: Hand moving about elbow

In PCLI, linear interpolation is performed on different parameters to CCLI, that is length and angle of each segment on the boundary curve. One form of PCLI is known as the method of interior angles. Applying PCLI to the foregoing example we get correct results because PCLI can overcome the rotation distortion which CCLI suffers from. In fact PCLI can achieve good results in many applications e.g. when dealing with local drawings in Figure 3. Figure 6 shows an example.

In some cases PCLI might transform closed polygons into open ones or cause intersections thus resulting in changing the polygon area in the intermediate shapes. We can derive an example of this happening from the two closed polygons as shown in Figure 7. Here the two neighbouring segments joining at  $P$  remain the same in length and the curve is concave and convex respectively at  $P$  in two key shapes (Mathematically this corresponds to the sign of the curvature

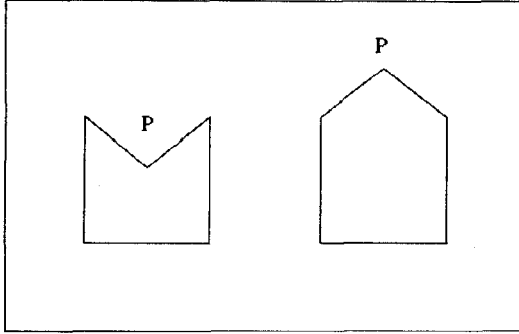


Figure 7: Two polygons with large changing in curvature value

changes at  $P$ ).

Since PCLI tends to preserve the length of two segments during the interpolation process, the resulting polygons will open up under PCLI interpolation. Closure can be guaranteed if the optimisation algorithm as suggested in [11] is used, but the resulting polygon area would vary incorrectly. Further investigation shows that the opening up problem arises not only when the sign of the curvature changes at  $P$ , but also might arise when the difference between the two corresponding curvature values is large even though both curvature values are of same sign. It is difficult to calculate how big the difference would have to be before the polygon starts to open up because this also depends on the length of the two neighbouring segments.

From this we can still get a loose criterion for an invalid condition for PCLI, i.e. calculating the curvature at each point on the polygon for two key shapes. If the sign of the curvature for any corresponding vertex changes then PCLI is invalid. Here, by curvature, we mean the vector whose origin is the vertex, direction is normal to the line joining the two vertices on either side, and length is approximately that of the radius of curvature for the circle which most nearly approximates the shape of the three vertices. If the polyline (polygon) is approximating a smooth curve the curvature is taken from the original curve directly.

Later we will show that PCLI is a good candidate for dynamic control of a skeleton, and is particularly suitable for an object with an articulated skeleton because PCLI naturally works in the same way as the skeleton of an articulated object. A human figure is a typical example of an articulated structure so PCLI applied to its skeleton can produce in-betweens without the opening up or intersection artefacts which could otherwise arise if PCLI were applied to the silhouette directly.

## 2.4 Calculation of the rotation angle between key polygons

This section gives a curvature formula which can be used to calculate the rotation angle between key polygons.

It has been shown that the curvature change along the boundary of the object's is independent of the object orientation [6], and the curvature  $\rho(u)$  at  $(x(u), y(u))$  is given as follows

$$\rho(u) = \frac{x''(u)y'(u) - y''(u)x'(u)}{\sqrt{x'(u)^2 + y'(u)^2}^3}. \quad (3)$$

The value of  $\rho(u)$  is invariant under the rotation which can be checked easily by substituting the following  $x(u)$  and  $y(u)$ ,

$$\begin{bmatrix} x(u) \\ y(u) \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \tilde{x}(u) \\ \tilde{y}(u) \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix}. \quad (4)$$

into equation (3), where  $a, b, \theta$  are constants.

Since the value of  $\rho(u)$  is invariant under the rotation, we can make use of this property to calculate the rotation angle between two key shapes as follows: first calculate the curvature of the polygon using the formula given in [3], then pick up some points with big curvature values as key points which characterise the main feature of the polygon, and with those key points we can form a so called feature polygon of the original polygon. We calculate the angles of each segment in the two feature polygons, i.e.  $\theta_{1i}$  and  $\theta_{2i}$  ( $i=1, \dots, M$ ), where  $M$  is the number of vertices contained in the feature polygon, then calculate the difference between the corresponding angle values  $\Delta\theta_i$ , and finally take the maximum difference value of them as the rotation angle  $\theta$  between two key feature polygons. This angle of rotation also measures the degree of rotation of the original polygons.

The foregoing, if combined together, can form a model for use in interpolant recommendation and acceptance judgement.

## 2.5 Structure of the model

The area assessment model applies to both shape and skeleton techniques which can be expressed by the following:

1. Calculate the curvature of the curve,
2. Calculate the rotation angle  $\theta$  between the two key shapes,

3. If  $\theta < \pi/6$  then recommend CCLI else recommend PCLI,
4. Use Length-preservation or Area-preservation criterion according to application,
5. Make the judgement for acceptance.

### 3 Dynamics

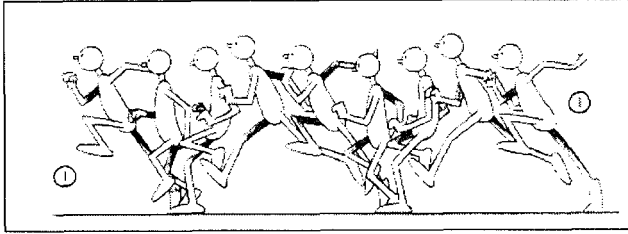


Figure 8: A correct run cycle in hand drawn animation

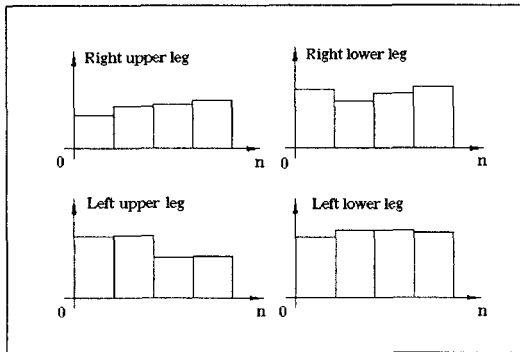


Figure 9: Angle variation of two legs for a run

At present our numerical criterion concerns the static shape area of deformed object only. It still cannot tell for sure whether these intermediate shapes are dynamically correct, but dynamic correctness of the movement is also a key factor to the success of animation.



Figure 10: A walk using angle constraints

From example (c) of star-skeleton [13] in Figure 2, despite the complexity of the star-skeleton representation extraction procedure, many people claim this result is correct because it uses a natural representation of a polygon, accounting for its important geometric features. Is this true? From the static aspect, the star-skeleton represents the structure of the object well, but from the dynamic aspect, it is unclear whether linear interpolation performed on the star-skeleton will produce correct results. The human figure in Figure 2 looks like it is running, and Figure 8 [14] shows the correct dynamics for a run used in hand drawn animation. In fact the dynamics of the two legs cannot be controlled correctly by linear angle interpolation on leg skeletons, because the angles of two leg skeletons vary non-linearly within one run step. Figure 9 shows the graph of angle against frame  $n$  which is drawn by measuring angles in the skeletons of the two legs in Figure 8. Possible solutions to this problem are interactive control to the interpolation process, which means more manual work, or controlling them procedurally as proposed in [16]. In the latter case the idea is to specify angle values for the upper and lower legs derived from the hand drawn walk cycle as angle constraints on PCLI when controlling the skeleton dynamics of the two legs. The walk type and speed are also controllable and Figure 10 shows an example of this approach. A similar idea can be applied to running and the corresponding angle values of Figure 9 can serve as its angle constraints.

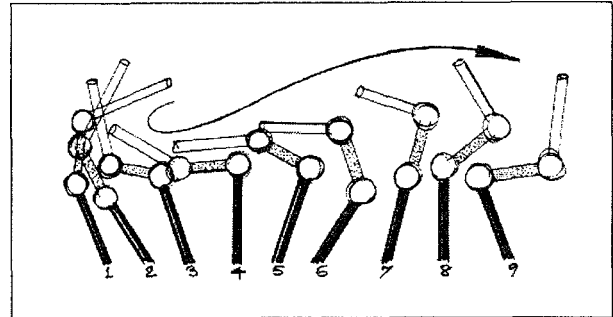


Figure 11: The movement of three sticks

The above example showed a dynamic control problem for an animated object. The problem also exists for inanimate objects. Figure 11 [4] shows the movement of three stick flexibly joined together. The angle between each stick would vary non-linearly and differently if graphed against time. Linear angle interpolation, as used in PCLI, cannot deal with this case either so we should use a different approach such as the one suggested already to solve the problem.

Nevertheless, since actions like walking and running for animate objects are represented in a fixed manner in hand drawn animation, those angle constraints characterise the dynamics of legs in such motions. Therefore they can be used for an objective assessment of morphing techniques when dealing with these cases. If we take the running figure (c) in Figure 2 as an example, where their angle interpolation is performed on the star-skeleton (especially those segments corresponding to legs) and is then in accordance with those angle constraints, then the result is correct, otherwise not. This kind of assessment can be extended to other movements like forward or backward facing walking figures, sneaks (creeping motion) etc. In a similar way that the area preserving criterion is a guiding principle, so these angle constraints serve to aid assessment of dynamical behaviour.

## 4 Timing

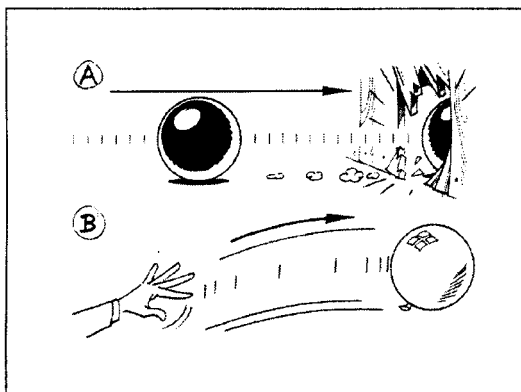


Figure 12: A cannonball and a balloon

Timing is another key factor in animation. Since timing is reflected by positions of a moving object in space, it is also related to the shape interpolation process. Like the dynamic control problem, we cannot guarantee correct timing even if we get the correct shape. Figure 12 [4] shows a cannonball and a balloon both represented by circles. A cannonball need a lot of force to start its moving and, once it moves, it takes a lot of force to make it stop. A balloon needs only a small force to move it, but air resistance quickly brings it to rest. In both examples a circle is being animated, but the timing of the movement can make it look heavy or light on the screen. The way an object behaves on the screen, and the effect of weight that

this gives, depends entirely on the spacing of the animation drawings and not on the drawing itself. It does not matter how beautifully drawn the cannonball is in the static sense, it does not look like a cannonball if it does not behave like one. The same applies to the balloon and indeed to any other object or character.

Timing in animation is an elusive subject. It only exists while the film is being projected, in the same way that a melody only exists while it is being played. We need good timing in animation, so that enough time is spent preparing the audience for something to happen (anticipation), then on the action itself, and then on the reaction to the action (follow-through). To judge this correctly depends upon an awareness of how the minds of the audience work. How quickly or how slowly do they react? How long will they take to assimilate an idea? How soon will they get bored? This requires a good knowledge of how the human mind reacts when being told a story. In animation, timing becomes a dangerous factor to try to formulate – something which works in one situation or in one mood may not work at all in another situation or mood. The only real criterion for timing is: if it works effectively on the screen it is good, if it does not, it is not.

Nevertheless, the experience of animators may provide a basic understanding of how timing in animation is ultimately based on timing in nature and how, from this starting point, it is possible to apply such a difficult and invisible concept to the maximum advantage in film animation. Despite these warnings we still think it is possible to make objective timing assessments in certain conditions. If we return to the cannonball and balloon. Suppose their trajectories are the same, so can be represented by  $x(t)$ ,  $y(t)$  in the  $X - Y$  plane (screen). The differential timing for the cannonball and balloon can be obtained by different sampling on  $x(t)$ ,  $y(t)$ , i.e.  $CTx(t_i)$ ,  $CTy(t_i)$  for the cannonball timing and  $BTx(t_i)$ ,  $BTy(t_i)$  for the balloon timing. With such timing functions the distance between each pair of adjacent timing sampling points is measurable and this can be used for objective timing assessment. At least we can use these functions to distinguish them. But it is hard to make a general timing assessment because timing is object dependent like the case of the cannonball and balloon, or application dependent like the different timing of rain can suggest different moods.



## 5 Movement continuity, path direction and continuity

Movement continuity, path direction and continuity are relevant factors in the assessment of morphing techniques [8], because timing and shape control are inevitably bound up together in certain 2D animation applications. A good way of showing this is in terms of moving reference points [10], although the use of this technique for specifying in-betweening problems is questionable. We imagine a reference point on the forehead of a character seen from the side. While continuing to look in his direction of motion the character takes a single step forwards, then stops. The reference point now describes a roughly straight-line trajectory (modified only by gait) with equal time points clustering near the beginning and end of the movement, reflecting the ease-in and ease-outs of the movement. The head shape does not alter so the relative positions of all reference points on the face will remain the same, yet the timing along the trajectory is not simple. If, instead the character turns to face the viewer without otherwise moving, again the reference point executes a complex trajectory following the change in the forehead silhouette, that is it models a shape change only. If the two movements are combined, the result is a third trajectory, different to first two, which encodes both shape changing and timing information.

This example so far has covered only smooth changes in movement (unless the step is very silly with the head bobbing up and down), and the curve can be modelled in terms of  $C^2$ -continuous B-splines which normally show the kinds of flowing paths we associate with animated figures (prior to their encountering walls and other obstacles, that is). However, if the movement is silly enough, there may be a  $C^0$ -continuous event in the trajectory, effectively discontinuous motion (where the head may stop and change direction). This is also the case with the forehead silhouette change. If the character has hair then as the head turns the hairline will come into view and the forehead silhouette will become associated with one of the hair boundaries, let us say the hairline. In this case, the reference point will have a  $C^0$  event at the moment it is incorporated into the hairline. So it can be seen that less than  $C^2$  continuity may arise at instantaneous points either from shape changes or from trajectory alone and when the two are combined they result in frequent events along the trajectory line. Where moving reference points only are used each of these events corresponds to an intervention by the animator as they are otherwise unpredictable without

specific 3D shape information or higher-level information about the movement. When it is recognised that control of both shape and movement in this way requires a large number of reference points it becomes clear that a great deal of intervention is needed to avoid violating the viewer's sense of the third dimension in unplanned ways.

This is not meant to be a criticism of the use of moving reference curves (although it is, in fact) but rather an example to show the pernicious effects of not having 3D information built in. Wholly 2D in-betweening techniques can always be made to fail on what look like innocent examples although some methods may fare better than others in particular cases of difficulty. Moving reference curves are beguiling because animators use them to specify in-betweening problems to in-betweeners (junior assistants) but the human in-betweener then goes on to use world knowledge to resolve what, to a computer, would still be a heavily under-specified problem. When used directly to specify in-betweening problems to an automatic in-betweener they are both difficult to specify (because so many aspects of the movement of a point are bundled together) and tedious to use (because so many are needed). Here they are more useful for illustrating problems rather than providing solutions.

## 6 Effects

Automatic methods for in-betweening rely on some degree of continuity in both geometry and time. However, many effects, e.g. fire, waterfalls, ripples on water, rain, etc are implemented either by blending between discontinuous frames or by putting up successive frames where there is no planned continuity between elements and correspondences are difficult to establish or are even non-existent [17]. In such cases these automatic methods will fail utterly.

A method for dealing with effects, in terms of a hierarchical model for the images of a sequence, has been proposed by us [17]. Here the hierarchy goes from static elements at the bottom (e.g. extent of fire base), then high-continuity elements (like flame elements, which change slowly or not at all in number but may take different orientations), then low-continuity elements (like the flames which break off from the body of the fire), then random elements (e.g. some parameters and connection curves - effectively "fixing up" the artwork at the end). The random elements predominately determine the superficial appearance of individual images while the more continuous element determine the character of the movement associated

with the effect. In hand animation most of these effects are made up using a few drawings permuted at random or cycled but consideration of the more continuous elements is necessary if, for example, a distinction is to be made between a large fire and a small one (or a large waterfall and a small one). Advantage can also be taken of awareness thresholds to hold an image for a few frames then swap it for another, which can slow down fire and make it seem like a larger conflagration.

Timing problems appear here too. When random or stochastic elements are used there is the risk of running into moire effects, or fairly subtle aliasing effects, where high-frequency elements like raindrop trajectories can lead to low-frequency effects if the trajectories bunch up regularly in the cycle (once is enough). Essentially care has to be taken to detect and avoid these problems within the stochastic mechanism on a case-by-case basis.

In dealing with effects in this (procedural) manner questions of shape (what do the individual images look like?) predominate over questions of timing (frames may be ordered in time but there is no obvious correspondence let alone continuity between elements in successive frames). Continuity considerations are only relevant at lower levels of the hierarchy where there is a relationship with the final image but this is not at all obvious. The point is that procedural methods look like other in-betweening methods at a sufficient level of abstraction, although they have their special problems too.

## 7 Conclusion

It is not enough to consider shape information only to assess morphing techniques for the purposes of animation. We need to take into account other factors like structure, dynamics, timing etc. A fully objective assessment of the quality of a morphing algorithm is quite impossible, because it depends on multiple factors like the way timing is related to subjective judgement, but some objective assessments are locally possible such as those we have presented in this paper. We should point out that our criteria are heuristics rather than formally based rules, yet are the sorts of rules that animators are taught.

Since our numerical criterion was derived from the principle of traditional animation, it is naturally applicable to the problems we might encounter in practical hand drawn animation. Some papers on 2D morphing techniques show examples of morphing from one abstract object to another. The results from these exam-

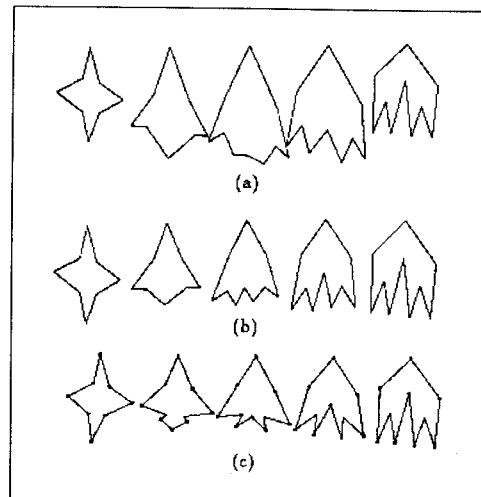


Figure 13: Comparative results of morphing techniques applied to a fancy shape

ples do not have any correspondence to ideas of what should happen, so are indistinguishable from magic, where the same phenomenon applies. In these cases any result could be regarded as correct, or for that matter incorrect, on arbitrary criteria. For the fancy shape shown in Figure 13 we cannot find its counterpart in our experience even for its static shape, to say nothing of its dynamic behaviour. Though the results in the figure are used for comparing different morphing techniques, we still cannot tell right from wrong so using any, let alone our, criterion to assess them is meaningless.

## References

- [1] N. Burtnyk and M. Wein. Computer generated key-frame animation. *Journal of Society for Motion Picture and Television Engineering*, 80:149–153, 1971.
- [2] N. Burtnyk and M. Wein. Interactive skeleton techniques for enhancing motion dynamics in key frame animation. *CACM*, 19:564–569, 1976.
- [3] Eli Goldstein and Craig Gotsman. Polygon morphing using a multiresolution representation. *Graphics Interface '95*, pages 247–254, 1995.
- [4] W. Harold. *Timing for Animation*. Focal Press Limited, London, 1981.
- [5] A Kaul and J Rossignac. Slويد-interpolating deformations: Construction and animation of pips. *Computers & Graphics*, 16:107–115, 1992.

- [6] Toshiyasu L. Kunii and Takao Maeda. On the silhouette cartoon animation. *Proc. of Computer Animation'96*, pages 110–117, 1996.
- [7] Alexander A. Pasko, Andrei V. Savchenko, and Vladimir V. Savchenko. Implicit curved polygons. *Technical Report 96-1-004, University of Aizu, Japan*, 1996.
- [8] John W. Patterson and P. J. Willis. Computer assisted animation: 2d or not 2d? *The Computer Journal*, 37:829–839, 1994.
- [9] V. Ranjan and A. Fournier. Matching and interpolation of shapes using unions of circles. *Computer Graphics Forum*, 15:C-129–C-142, 1996.
- [10] W. T. Reeves. Inbetweening for computer animation utilizing moving point constraints. *Computer Graphics*, 15:263–269, 1981.
- [11] T. W. Sederberg, P. Gao, G. Wang, and H. Mu. 2d shape blending: An intrinsic solution to the vertex path problem. *Computer Graphics*, 27:15–18, 1993.
- [12] T. W. Sederberg and E. Greenwood. A physically based approach to 2d shape blending. *Computer Graphics*, 26:25–34, 1992.
- [13] M. Shapira and A. Rappoport. Shape blending using the star-skeleton representations. *IEEE Computer Graphics and Applications*, 19:44–50, 1995.
- [14] T White. *The animator's book*. Watson-Guptill, New York, 1986.
- [15] Jinhui Yu. Inbetweening for computer animation using polar coordinate linear interpolation. *CS Report Series, CSC 90/R23, University of Glasgow, UK*, 1990.
- [16] Jinhui Yu. A walk model for computer-aided character animation. *CS Report Series, CSC 90/R30, University of Glasgow, UK*, 1990.
- [17] Jinhui Yu and John W. Patterson. A fire model for 2d computer animation. *Computer Animation and Simulation'96, SpringerComputerScience EG*, pages 49–60, 1996.