A FINITE DIFFERENCE ALGORITHM
FOR ACHIEVING NATURALISTIC ANIMATION

Don Herbison-Evans

Technical Report 140

January 1979

BASSER DEPARTMENT OF COMPUTER SCIENCE

ISBN 0 909203 87 3
THE UNIVERSITY OF SYDNEY
N.S.W. AUSTRALIA 2006
A FINITE DIFFERENCE ALGORITHM
FOR ACHIEVING NATURALISTIC ANIMATION

Don Herbison-Evans

Technical Report 140

January 1979

ISBN 0 909203 87 3
In animation, it is often necessary to vary some geometric parameter (e.g., angle, position) over a series of frames to give some prespecified total change to that parameter. If the frames over which the action is required are $s$ to $s+n$, and if the total change required of the parameter is $Q$, and if the increment of that parameter between frames $f$ and $f+1$ is $q(f)$, then a constant velocity solution to the animation (e.g., (2)) is to let

$$q(f) = \frac{Q}{n}$$

This solution has the disadvantage that infinite accelerations appear to occur at the start and the end of the motion. The eye is surprisingly sensitive to sudden changes in apparent velocity. This parallels, in the time domain, the sensitivity of the eye to sudden changes in the spatial derivative of grey level: Mach bands (1).

If it is required that the motion start at rest, and that it accelerate throughout the sequence of $n$ frames at a constant rate, then the increments $q(f)$ must increase uniformly:

$$q(f) = p \cdot (f-s)$$

where $p$ is some constant increment of velocity. The value of $p$ can be derived from

$$Q = \sum_{f=s}^{f=s+n} p \cdot (f-s) = p \cdot n(n+1)/2$$

i.e., $p = 2Q/(n^2+n)$

The same increment can be used for decelerating motion, starting with $n$ such increments for the first frame, $n-1$ for the second, etc.

A more natural form of movement starts and ends at rest. This has previously been achieved either by interactive specification (e.g., (3)) or using trigonometric functions (e.g., (4)). If, instead, this is implemented as a period of constant acceleration followed by an equal period of constant deceleration, this may be interpreted in terms of constant (but opposite) forces or torques being applied during the two halves of the motion. It can be implemented in a similar manner to the
constantly accelerating motion above, and if \( n \) is even, then the velocity increment is
\[
p = \frac{4Q}{(n^2+2n)}
\]
The peak value of the parameter increment is
\[
q(s+n/2) = np/2
\]
and this is repeated once at the halfway point, before beginning the decrement phase.

If \( n \) is odd, then
\[
p = \frac{4Q}{(n^2+2n+1)}
\]
The peak increment is
\[
q(s+(n+1)/2) = (n+1)p/2
\]
and is used only once.

After \( p \) has been calculated for any of these cases, their implementation involves only adding or subtracting \( p \) from the current value of \( q \), which is in turn added to the parameter being controlled. Thus it is well suited to implementation on a mini or intelligent graphics satellite computer.

References


