Introduction to Animation

Lecture 13

© Datafusion Systems 1999

Animation

- Animation introduces a dynamic element into an otherwise static presentation.
- Animation does not necessarily imply spatial motion.
- Any attribute of a scene/image that changes results in animation, including:
  - Colour / Transparency
  - Texture
  - Shape variations

© Datafusion Systems 1999

Reasons To Animate

- Reasons to animate:
  - Looks cool
  - Time varying data
  - Time varying simulation
  - Dynamic spatial visualisation techniques

© Datafusion Systems 1999

Time Varying Data

- In simple cases where a single datum is collected at regular time intervals, a 2D graph may be sufficient (one axis being time).
- In other cases entire data sets will be sampled at different points in time.
  - Typically each data set is meaningful alone and may be presented as a 2D surface or 3D volume.
  - Time can be used as a visualisation dimension.

© Datafusion Systems 1999

What Can Be Animated ?

- Camera
  - Motion of a camera around an object
  - Provides spatial context in a non-interactive environment.
- Lighting
  - Lighting angle affects shading and shadows
  - Enhances interpretation

© Datafusion Systems 1999

What Can Be Animated ? ...

- Objects
  - Images (movie)
  - Surface colour
  - Slice planes
  - Position
  - Deformation
- Objects include *meshes* and *particles*

© Datafusion Systems 1999

Animation Techniques

- There are many techniques for animation. The appropriate class of technique is dictated by context.
- Common classes are:
  - Simple time-stepping though data
  - Key-frame Animation
  - Simulation
  - Palette Animation

© Datafusion Systems 1999

Simple Temporal Animation

© Datafusion Systems 1999

Simple Temporal Animation

- Assume we have a dataset with a time varying component
- Determine which attributes of a visualisation are time dependent
- Let the time varying data be directly visualized
  - eg as colour, vector position, height etc
- Develop a visualisation for a single time step

© Datafusion Systems 1999

Simple Algorithm

- For each time-step in data:
  - Load data for time-step
  - Render visualisation
  - Create image file of scene
- The image files should be numbered sequentially and with sufficient leading zeros to accommodate the last frame, eg `image001.tiff`
- Place image files in a separate directory

© Datafusion Systems 1999

Simple Algorithm Results

- The resulting set of images may be played sequentially as an animation.
- Issues:
  - The time steps between data samples may not be uniform.
  - There may be large jumps between samples
  - If there are few time-steps, the result cannot usually be displayed as a "real-time" animation.

© Datafusion Systems 1999

Key-frame Animation

© Datafusion Systems 1999

Key-frame Animation

- This technique relies on a series of data sets or object locations at different points in time to guide the action.
- These key animation control sets are termed *key-frames*.
- Key-frames may contain:
  - real scientific data (eg each time-step in the previous section)
  - a set of camera/object position.

© Datafusion Systems 1999

Key-frames

- A *key-frame* is *not* just an image frame at a particular point in time, it is the entire definition of the system at that time.
- Relatively few key-frames are created. If rendered and played the result would be jerky motion (as may occur with the simple algorithm previously).
- Key frames are used to generate many intermediate frames by interpolation.

© Datafusion Systems 1999

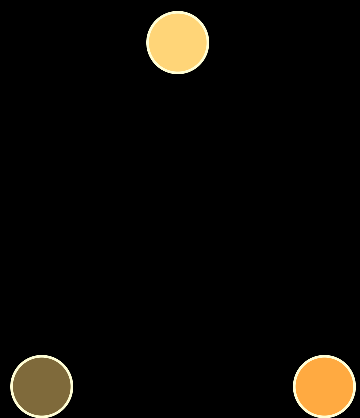
Key-frame Terminology

- `Frame` : A single image in an animation/ movie or the state of an animation system at point in time.
- `Key-frame` : an animation system state that acts as an interpolation control point.
- `Inbetweening/tweening` : generating the intermediate frames between two key-frames.

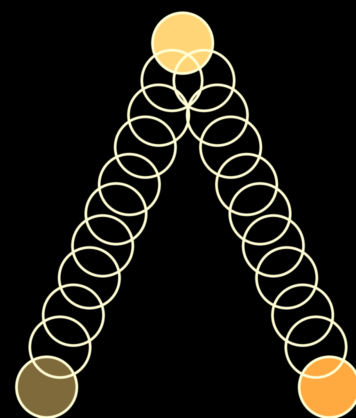
© Datafusion Systems 1999

Key-frame Animation

Example: Ball thrown up in air



3 key-frames



Linear interpolation is not always appropriate

Computer generates intermediate frames

© Datafusion Systems 1999

Key frame Interpolation

- Linear interpolation (*lerping*)
  - Simplest and often safest technique in sci vis.
  - Generates continuous motion but not continuous derivatives.
- Splines (a curve defined by control points often a piecewise cubic polynomial)
  - Often preferred for smooth motion
- Non-scientific parts of the animation may require gradual velocity changes.

© Datafusion Systems 1999

Simulations

© Datafusion Systems 1999

Simulations

- Instead of having real data sets sampled at time intervals, the time varying aspect of a visualisation may be computed.
- System state at a given time may be calculated from a set of initial conditions and constraints.
- This leads into the areas of Kinematics and Dynamics

© Datafusion Systems 1999

Kinematics and Dynamics

- Kinematics (Forward)
  - Position and velocity of points specified
  - Simple calculation generates data for each time-step.
- Dynamics (Forward)
  - Physical laws of Kinematics
    - Newton's laws of motion,
    - Euler-Lagrange fluid laws etc
  - Particle motion is a result of forces acting upon it.

© Datafusion Systems 1999

Inverse Kinematics / Dynamics

- Inverse Kinematics
  - Calculate the constant velocity required for an object to reach point *b* at time *t*.
- Inverse Dynamics
  - Calculate the forces required to act upon an object for it to reach point *b* at time *t*.
- These techniques are used to animates mechanical systems and human figures.

© Datafusion Systems 1999

Inverse Kinematics / Dynamics…

- Inverse K/D may not have unique solutions, therefore constraints are often added to produce unique solutions.
  - *Constraints based physical modelling.*
- Constraints may be
  - physical (eg axis of motion)
  - abstract (eg minimize potential energy)
- Constraints may be dynamic as in a spring system.

© Datafusion Systems 1999

Particle Advection

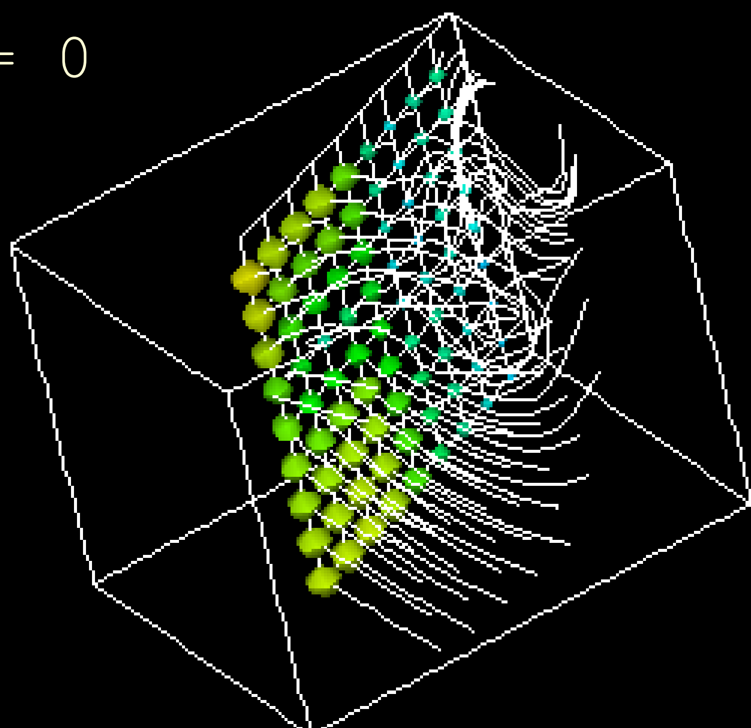
- Simple example of kinematics
- Use of indirect technique to display vector data
- Rather than display vectors directly, a *massless* particle travels through the vector field (volume), its motion being directed by the vectors.
- Animation is a series of time-steps at user specified temporal resolution.

© Datafusion Systems 1999

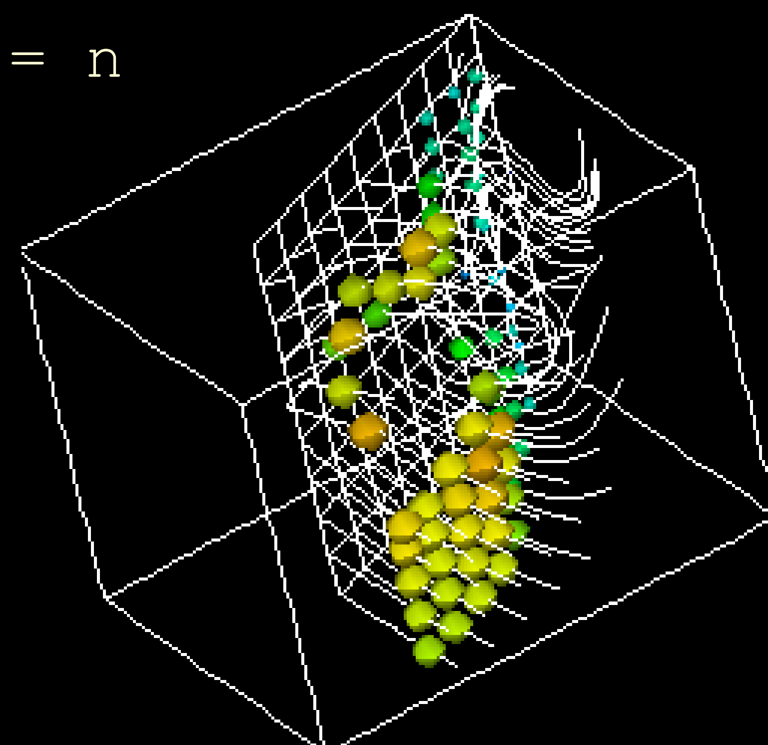
Particle Advection…

Massless particles flowing though a velocity field

$t = 0$



$t = n$



© Datafusion Systems 1999

Animation Issues

© Datafusion Systems 1999

Animation Issues

- Animations should enhance interpretation not just look fancy.
- If there are only a few images in the animation then each should be held for long enough to enable the salient points to be studied. (A voiceover helps determine length).
- Movies may run animations twice, slow for detailed study and once fast to see changes.

© Datafusion Systems 1999

Presentation Methods

- The decision to animate increases the base requirements for presentation
- When the number of time-steps is very low (say five) they may be more easily displayed as a series of static images.
- Possible solutions:
  – Video
  – Digital Movie Format (Mpeg/quicktime etc)

Bandwidth is still an issue for computer based movies.

© Datafusion Systems 1999

Animation Context

- Animation context is *very* important.
- It should be clear to the viewer precisely what is being animated.
- Ideally each image frame should contain enough information to be viewed (at least partly) out of context.
- **Each frame should contain a legend and time/state information** as appropriate.

© Datafusion Systems 1999

Animation Context…

- A title should introduce the animation.
- Large fonted text describing the animation should preface any animation (Generally at least 28pt).
- Text should remain visible for around twice the time it takes to read it very slowly.
- If possible a voice-over should accompany animation.

© Datafusion Systems 1999

Animation Gotchas

- Temporally constant scales
  - Packages like AVS by default autoscale colourmaps etc to the available data range at *each* time-step.
  - These scales need to be fixed appropriately for the entire animation.
- Disable any visualisation package modes such as auto-normalization.
- Disable any screen-savers when generating animations over an extended period.

© Datafusion Systems 1999

Temporal Aliasing

- Generally not a problem in simple dataset visualisation unless time-sampled data has an aliasing problem
- Related to sampling theory (Nyquist limits)
- An element of an animation that is changing too fast compared to the animation frame rate
- Common example, wheel spokes appearing to move backwards on a vehicle

© Datafusion Systems 1999