Computer Animation: Art, Science and Criticism

Tom Ellman

Harry Roseman

Lecture 3
Conventional Key Frame Animation

• Master draws “Extremes”.

• Apprentice draws “Inbetweens”.
Extremes

\[ t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \]
Extremes and Inbetweens
Key Frame Computer Animation

- Human artist sets key values.

- Computer uses mathematical *interpolation* to compute intermediate values.
Interpolation

Generate a curve that goes through a given series of discrete points.
Continuous Time v. Discrete Time

• Animation:
  – Time is continuous.
  – Keys may be set at arbitrary real times.
  – Interpolated values are defined at all real time values.

• Rendering:
  – Time is discrete.
  – Frames are generated only at selected time instants.
Setting Keys

• Auto Key:
  – Create a new key each time the user changes the value of a variable that already has a key.

• Key Selected:
  – Create a key using the current values in the selected channels.

• Key All:
  – Create a key using the current values in all channels.

• Setting keys in Graph Editor:
  – Insert key: Put new key on existing curve.
  – Add key: Put new key off or on existing curve.
Ambiguity of Interpolation

- Suppose we are given a series of discrete points.
- An infinite number of curves can be drawn to go through all the points.
- How can computer decide which curve is the right one?
- The human animator must supply additional information.
Classes of Curves

Constant: \( p(t) = a_0 \)

Linear: \( p(t) = a_1 t + a_0 \)

Quadratic: \( p(t) = a_2 t^2 + a_1 t + a_0 \)

Cubic: \( p(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0 \)
Continuity and Smoothness

Continuous

Smooth

Discontinuous

Non-Smooth
Piecewise Cubic Polynomials

Cubic Polynomial

Cubic Polynomial

Cubic Polynomial

Cubic Polynomial
Cubic Polynomials

\[ p(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0 \]

- A cubic polynomial \( p(t) \) has four coefficients.
- We can define a cubic polynomial by specifying \( a_0, a_1, a_2 \) & \( a_3 \).
- Alternatively, we can specify any four quantities from which \( a_0, a_1, a_2 \) & \( a_3 \) can be computed.
**Hermite Curves**

- User specifies value of curve $p(t)$ at each interpolation point.
- User also specifies the slope (tangent) $p'(t)$ at each interpolation point.
Value and Slope of Cubic Polynomials

Value: \[ p(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0 \]

Slope: \[ p'(t) = 3a_3 t^2 + 2a_2 t + a_1 \]

Calculus!
Specifying Tangents in Maya

• Flat tangent:
  – Tangent is horizontal.

• Stepped:
  – Curve is flat between keys. (Discontinuity)

• Linear tangents:
  – Tangent points toward next interpolation point.

• Spline:
  – Tangent is average of prior and current linear tangents.

• Clamped:
  – Like spline, but linear when two keys are close to each other on the time line.

• Plateau:
  – Keeps curve inbetween minimum and maximum values of keys.
Flat Tangent

$p_{n-1}$  $p_n$  $p_{n+1}$
demo-03-02a-flat-tangents.mb
Linear Tangent

$p_{n-1}$ $p_n$ $p_{n+1}$
(Catmull-Rom) Spline Tangent

\[ p_{n-1} \rightarrow \text{Prior Linear Tangent} \rightarrow p_n \rightarrow \text{Current Linear Tangent} \rightarrow p_{n+1} \]

Average of Prior and Current Linear Tangents
Plateau

Spline

demo-03-02d-plateau-tangents.mb
Fixed Tangents

• User controls tangent by manipulating handles.
• Unified Tangents v. Broken Tangents:
  – Use (Graph Editor) UnifyTangents and BreakTangents buttons to unify and break tangents.
  – Unified: Incoming and outgoing tangents are the same.
  – Broken: Incoming and outgoing tangents are different.
• Unweighted Tangents v. Weighted Tangents:
  – Use (Graph Editor) Curve-WeightedTangents and Curve-UnweightedTangents menu items to turn on and off tangent weights.
  – Use (Graph Editor) FreeTangentWeight and LockTangentWeight buttons to free and lock tangent weights.
  – Unweighted: Tangent has fixed length.
  – Weighted: Tangent has user-controlled length.
  – Length determines how long tangent’s influence lasts.
Interpolating Rotations

• Euler Angle Interpolation:
  – Maya interpolates Euler angles in the same way as it interpolates other kinds of values.
  – An object can make multiple revolutions in moving from one key orientation to another.

• Quaternion:
  – Maya finds the shortest, most direct rotation to get from one key orientation to another.
  – An object never makes rotates more than 180° between two key orientations.

• Graph Editor menu item:
  – Curves-ChangeRotationInterp.

demo-03-03-rotate.mb
Describing Rotation with Euler Angles
Describing Rotation in Terms of Quaternions (Axis and Angle)

A direction vector $\mathbf{d}$ indicates the axis of rotation.

A scalar $\theta$ indicates the angle of rotation.

Describes only the final orientation resulting from a rotation – not the number of turns taken to get there.
Independent v. Synchronized

• Independent Euler:
  – User sets keys separately for x, y & z Euler angles.
  – E.g., if Euler-x has a key value at time t, Euler-y and Euler-z may or may not have key values at time t.

• Synchronized Euler:
  – User sets keys together for x, y & z Euler angles.
  – E.g., if Euler-x has a key value at time t, Euler-y and Euler-z will also have key values at time t.

• Graph Editor menu item:
  – Curves-ChangeRotationInterp.
Extrapolation to Infinity

• What happens after the last key?

• What happens before the first key?
Extrapolation to Infinity

- **Constant:**
  - Maintain the last key value.

- **Linear:**
  - Maintain the rate of change at the last key value

- **Cycle:**
  - Repeat the curve interpolated between the first and last keys.

- **Cycle with Offset:**
  - Repeat the rate of change between the first and last keys.

- **Oscillate:**
  - Repeat the curve interpolated between the first and last keys.
  - Make time run forward, backward, forward, backward, etc.

- **Graph Editor Menu Items:**
  - Curves-PreInfinity and Curves-PostInfinity.
  - View-Infinity.
demo-03-04d-cyclic-offset-infinity.mb
Editing Key Sequences

• Scale:
  – Stretch (slow down) a sequence.
  – Compress (speed up) a sequence.

• Cut and Paste:
  – Duplicate a sequence.
  – May or may not connect with prior sequence.

• Use Edit menu in either the Graph Editor or the Dope Sheet.

demo-03-05-bounce.mb
Paste Absolute v. Paste Relative

- **Absolute:**
  - Keys are pasted into new range without adjustment.
  - Analogous to Cycle Post-Infinity option.

- **Relative:**
  - Keys are adjusted to connect with existing curves.
  - Analogous to Cycle With Offset Post Infinity Option.

- Controlled by “Connect” checkbox in Edit-Paste[] menu item of Graph Editor or Dope Sheet.
Duration of time interval: N-1  Number of time values: N

Duration of time interval: N  Number of time values: N+1

Cut and paste usually works best when numbering frames: 0, 1, 2 … N. Replicated motions cover intervals: (0…N), (N…2N), (2N…3N) … etc.
Squash Deformer
Creating Squash Deformer

• Select sphere.
• Activate animation menu set.
• Invoke CreateDeformers-Nonlinear-Squash.
Applying the Squash Deformer

• Use Outliner to select squash handle.
• Invoke Modify-TransformationTools- ShowManipulatorTool.
• Use Move tool to change center of squashing.
• Handles allow:
  – Changing amount of squash.
  – Changing region of space impacted by squash.
  – Introducing asymmetry into squash.
• Input section of Channel box shows squash parameters that can be animated.
Bend Deformer
Jump Study Series

• We animate a jumping ball using the following channels: \( translateY \), (squash deformer) \( factor \), (bend deformer) \( curvature \) and \( translateX \).  
• We build up the animation gradually adding one new track at a time.  
• We examine the effect of each track by itself, and as it is combined with the others.  
• We synchronize keys on different tracks by comparing curves in the Graph View.
Animating \textit{translateY}

- The ball remains still for a period.
- Then it suddenly moves up into the air for no apparent reason.
- It seems to follow a “ballistic trajectory”.
- Then it hits the ground and stops instantaneously.
Animating `translateY` and (Squash Deformer) `factor`

- The squash `factor` goes down slowly, and then comes back up quickly.
- The squash `factor` channel stops rising at the instant the `translateY` channel begins to rise, and the ball rises into the air.
- Momentum is transferred from the squash `factor` channel to the `translateY` channel.
- Later squash factor starts to drop at the instant `translateY` channel stops dropping, when the ball hits the ground.
- Momentum is transferred back from the `translateY` channel to the squash `factor` channel.
Animat**ing** \textit{translate}Y, (Squash Deformer) \textit{factor} and (Bend Deformer) \textit{Curvature}

- The ball bends to the right (positive bend \textit{curvature}) as it squashes down (decreasing squash \textit{factor}).
- The ball begins unbending (decreasing bend \textit{curvature}) at the instant it starts unsquashing (rising squash \textit{factor}).
- The ball is standing straight up (zero bend \textit{curvature}) at the top of the jump.
- The second of half of the animation is not quite symmetric with respect to the first half.
- Should it be symmetric?
Animating `translateY`, (Squash Deformer) `factor` and `translateX`

- The `translateX` channel is flat while the ball is in contact with the ground, before the jump.
- The `translateX` channel is a straight line (increasing) while the ball is in the air, indicating uniform horizontal velocity.
- The `translateX` channel is flat again while the ball is in contact with the ground, after the jump.
Animating $\text{translateY}$, (Squash Deformer) factor and (Bend Deformer) curvature and $\text{translateX}$

- Now we combine all the channels into a single animation.
- The channels seemed to be correct when we viewed them separately.
- Do they work together as a group?
- Perhaps some adjustments are necessary.
Adding a Bit of Mid-Flight Squash

• We want the ball to “voluntarily” squash itself while it goes through the peak of its flight.
• We do this by adding a squash deform factor key at the time of greatest altitude.
• The squash factor is low at this point, indicating a vertically compressed ball.
• It makes the ball seem to float a little longer or higher than it would otherwise.
Exercise

• Download demos-03.zip and unzip to desktop.
• Open scene file: jump-study-04.mb
• Modify the animation:
  – To make the ball seem more rigid.
  – To make the ball seem less rigid.
  – To make the ball seem more eager.
  – To make the ball seem less eager.
Path Animation

• Allows separate description of motion path and time/speed of traversal.
• Allows an object to move forward and/or backward on a path.
• Makes it easier to avoid collisions with motionless objects.
Path Animation

Path Controlled Translation

Path Controlled Translation & Rotation
Varying Speed Along the Path

\[ \mathbf{c}(t) \]

Speeding up along path. Slowing down along path.
Making a Motion Path

- Use Create-EPCurveTool to make a NURBs curve.
- Create a “flying” shape (e.g., a plane, bird or fish).
- Note the “Front” and “Up” axies of the shape.
- Select the shape and then the curve.
- Active the Animation menu set.
- Invoke : Animate-MotionPath-AttachToMotionPath[] and select the “Front” and “Up” axies and confirm.
- Play the animation and notice that the object follows the path with its “Front” axis forward and its “Up” axis roughly upward.
Front and Up Axes of an Object

- Front axis is Y.
- Up axis is Z.
demo-03-06a-motion-path-example.mb
**Animating Motion Along a Path**

- Select the flying object; Go to the channel box; Find the Inputs section and Select motionPath1.

- Notice the U Value channel? This controls progress along the path: Zero is the start of the path; One is the end of the path.

- Move the time slider to a frame 1/3 through the animation; Select “ShowManipulatorTool”; Go to Perspective View and Click drag the yellow diamond on the flying object with LMB to move it 2/3 along the path. Select the U Value channel with LMB and do RMB KeySelected.

- Move the time slider to a frame 2/3 through the animation; Select “ShowManipulatorTool”; Go to Perspective View and Click drag the yellow diamond on the flying object with LMB to move it 1/3 along the path. Select the U Value channel with LMB and do RMB KeySelected.

- Play the animation and notice that the object goes forward and backward along the path.
The numerical markers indicate the frame numbers and path positions of keys. They can be selected and moved along the path with the move tool.
The animation curve shows how the “flying” object progresses forward and backward along the path.
Motion Path Options

• **Follow:**
  – Control rotation as well as translation.
  – Specify *Front* and *Up* axies.
  – Separately animate *Front*, *Up* and *Side* twist to vary rotation from strict *Follow* mode.

• **Bank:**
  – Rotates object the motion path tangent.
  – Points the objects’s *Up* vector toward the inside of a curve.

• Select these options when attaching an object to a motion path, or by opening the motion path in the attribute editor after it’s created.
demo-03-07-motion-path-banking.mb
Up Vector Anomaly

- What happens to the up vector when the path is exactly vertical?
- All rotations around the path tangent leave the up vector exactly horizontal.
- When the path goes through a point or segment at which the tangent is vertical, the object may rotate discontinuously.
- Solution: Avoid vertical motion paths.
Constraining the Up Vector

- The “Up” direction of the flying object may be constrained to point at another object.
- Select “Object Up” as the “World Up Type” in the options box when creating the object path.
- Enter the name of an object in the scene in the “World Up Object” slot.
- (E.g., Create a locator object for this purpose.)
demo-03-10-object-up.mb

Object on a motion path with its “Up” direction constrained to point at a locator object.
Motion Path Parameterization

• Parametric Distance:
  – Linear animation curve corresponds to a uniform rate of motion along the physical length of the motion path.

• Parametric Length:
  – Linear animation curve corresponds to a uniform rate of motion along the $u$-parameter space of the motion path.

demo-03-11-motion-path-parameterization.mb
Layered Motion Paths

• Attach spaceship to circular motionPath1.
• Attach circular path to linear motionPath2.
• Move simultaneously along both paths:
  – Traverse circular path 4 times from frame 0 to frame 800.
  – Traverse linear path 1 time from frame 0 to frame 800.
• Overall motion of the spaceship is a spiral.
• Parent a particle emitter to the spaceship.
• Particles trace out the ship’s spiral path.
Spaceship goes around circular path four times while circular path moves along a straight line.

demo-03-12-layered-paths.mb
Successive Motion Paths

- Attach spaceship to circular motion path.
- Attach circular path to linear motion path.
  - Traverse circular path 1 time from frame 0 to frame 400.
  - Traverse linear path 1 time from frame 400 to frame 800.
- Overall motion of the spaceship is a first a circle and then a straight line.
- Parent a particle emitter to the spaceship.
- Particles trace out the ship’s circular/linear path.
First spaceship moves around circular path. Then circular path moves along straight line.

demo-03-12-successive-paths.mb