Johns Hopkins
Engineering for Professionals

605.767 Applied Computer Graphics

Brian Russin

Module 12B
Animation Sequences

Animation Clip

- Pre-defined set of poses associated with time
  - Every frame's pose is defined
    - Memory-intensive
  - Key poses at specific times
    - Intermediate frames interpolated
    - Compute-intensive, but generally preferred
- Applies to entire model, or portions of a model

Animation Clip (cont.)

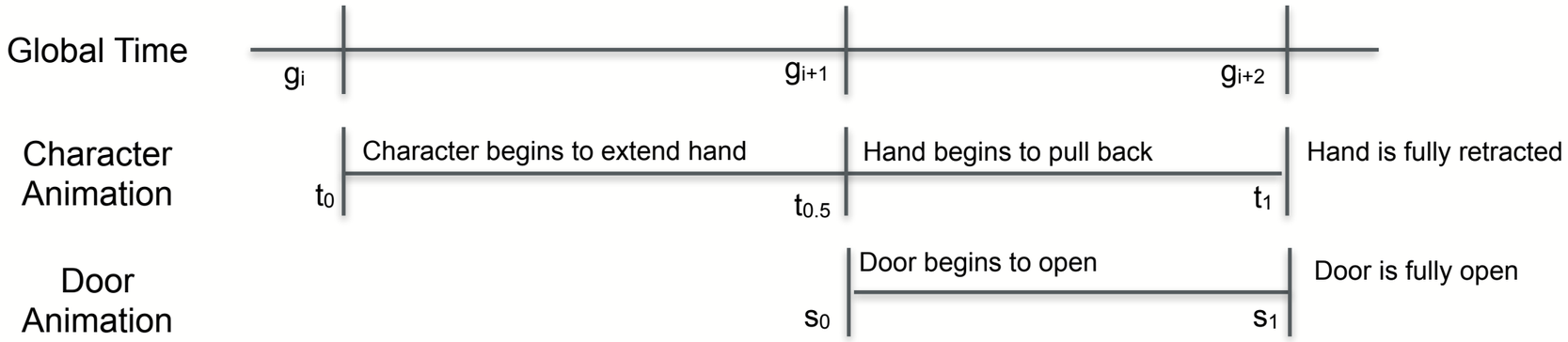
- Playing a clip
  - Play 1, n, or **infinite** number or times
    - Infinite means indefinite; until application received an interrupt
- Video Game Animation Types
  - NIS - Non-Interactive Sequence
    - e.g. character movements
  - IGC - In-Game Cinematic
    - Uses game assets
    - Longer that NIS (typically)
    - Cannot be interrupted
  - FMV - Full Motion Video
    - Pre-rendered and stored as video file

Animation Timeline

- Timeline defined from 0 to 1
  - Does not equate to 0-seconds to 1-second
  - Any interval would work, but 0-1 is simple
    - 0-1 easily scales to any time interval
    - Sync two animations (walking and running) easily
  - Some **animations repeat** (or **loop**)
    - **Key frames** at 0 and 1 are **equal**
- Key frames distributed throughout
  - Not necessarily uniformly distributed
- Essential to use **floating point time**
  - e.g. *std::chrono::high_resolution_clock*
  - Interpolate key frames for intermediate poses (called **samples**)
  - Loss of precision is acceptable in most cases

Synchronizing Time

- Start a local timeline at a global time
  - Trivial - managed by the parent process
- Synchronize two or more animation clips
  - Animation timelines to begin and end with others
  - e.g. Character opens a door, and the car door opens
    - Character reaches hand for car door, then pulls the hand back
    - Car door opens as the character's hand pulls back
    - Door opening may need to scale to match the hand's motion
  - Chaining animations
    - Controlled my a parent process to align the local timelines with the global timeline

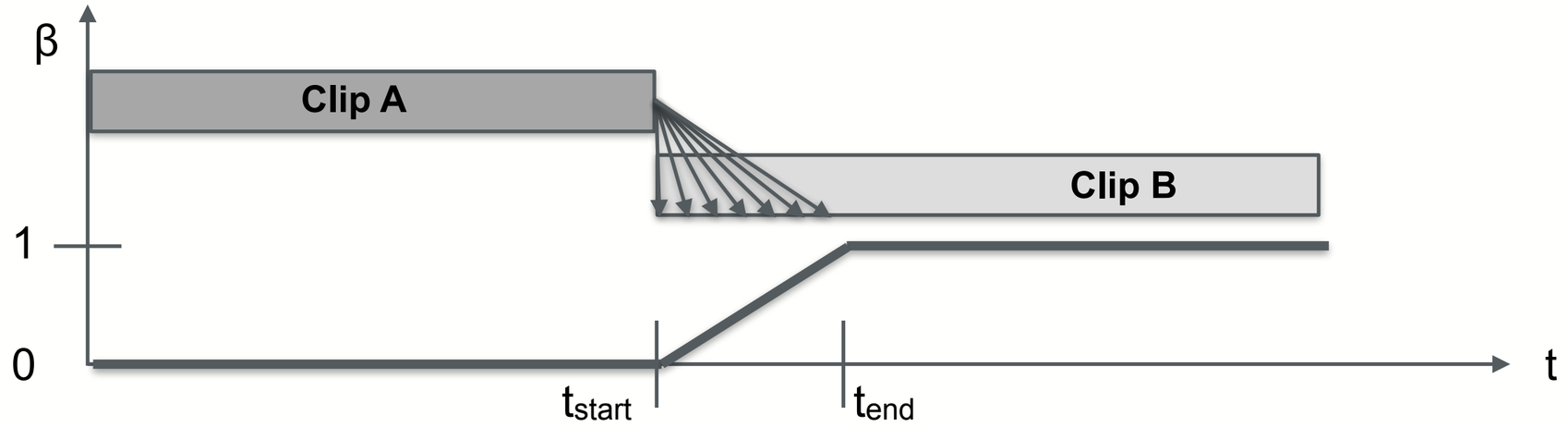


Animation Blending

- Interpolating between key frames
- More than one animation contributes to the final pose
  - Same technique as interpolation
  - Examples:
    - Injured-Walk blends walk animation with an injured pose
    - Player interrupts an animation
      - Application smoothly blends the end of the active animation with beginning of the triggered animation
        - Other options
          - wait until current animation ends - may be too long
          - abruptly end the current animation - may be too jarring

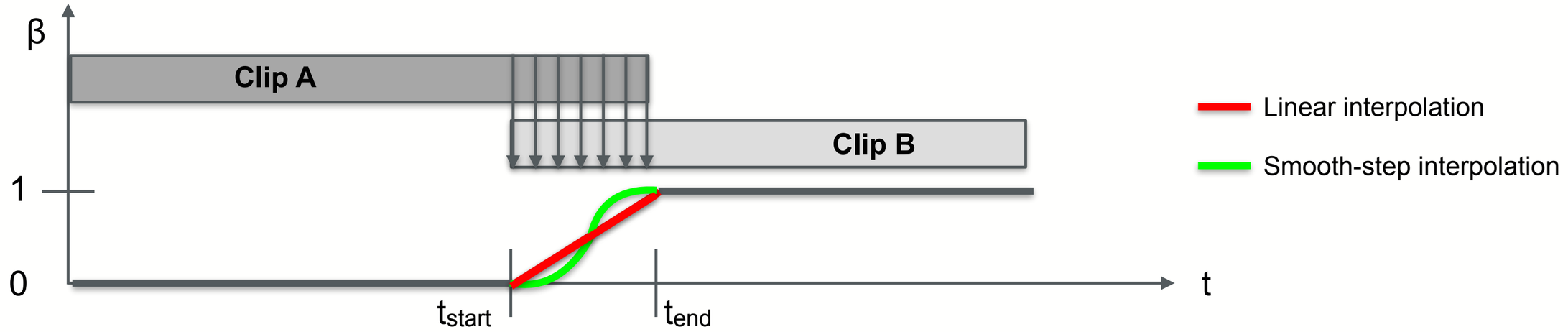
Animation Blending Techniques

- Blending Poses
  - Transformation matrices cannot easily be interpolated
  - Better to use the raw scale, rotation (as a quaternion), and translation values
- Freeze blending
  - Blend final pose of one clip with some portion of the next clip



Animation Blending Techniques (cont.)

- Overlap two animation clips
  - Ensures C0 continuity
  - Linear interpolation
    - $c_t = c_A + t(c_B - c_A), t: [0, 1]$
  - Smooth-step interpolation
    - $c_t = c_A + (3t^2 - 2t^3)(c_B - c_A), t: [0, 1]$



Additive Animation Blending

- Add the difference between two animations
  - First compute the difference between two clips
    - Reference clip (R): Running animation
    - Source clip (S): Tired running animation (S)
    - Difference clip (D): The change in poses to get from R to S
      - D = S - R or S = D + R
        - SRT differences in the key poses
- Difference clips
  - Can be applied to other clips as well
    - Beware: can cause incorrect transformations, like over-rotation of a joint
  - Small poses like the character looking to the left or right while moving

Procedural Animation

- Pose based on user interaction or the environment
  - Movement or pose added to a base pose
    - e.g. character looks left as controller moves left
    - e.g. car chassis oriented on the terrain
  - Poses can be precomputed, but not the time of execution
- Inverse kinematics (IK)
  - Intermediate joints positioned based on the desired leaf position
    - End effector: the target joint's position
    - e.g. Character reaches for a door knob, but the character's exact position, and the position of the door knob are unknown
  - Best if model is within a distance threshold
  - Error minimization problem
    - Has one, many, or no solution

Procedural Animation (cont.)

- Rag doll effect
  - Mimics a limp character model
    - Model’s appendages can be in any position
  - Incorporates some physics into the pose
    - e.g. Each joint has a physics-enabled bounding sphere

Constraints

- Joint transformation should be reasonable
  - Restricted articulation
  - e.g. A knee only bends (rotates) so far
  - Joint information can have its constraints as member data
- Inverse Kinematics
- Two or more relative models
  - Can use a special joint
  - e.g. objects inside or held by another
  - e.g. two models standing and facing each other