Realtime Computer Graphics on GPUs

Animation

Jan Kolomazník

Department of Software and Computer Science Education
Faculty of Mathematics and Physics
Charles University in Prague

Computer Graphics
Charles University

Vertex Animation

What is Vertex Animation?

- Vertex animation involves the manipulation of individual vertices to create movement and deformation of 3D models.
- Typically used for animating complex deformations and morphing effects.
- Unlike skeletal animation, vertex animation directly modifies the positions of vertices.

Key Techniques in Vertex Animation

- **Keyframe Interpolation**: Define vertex positions at key points in time and interpolate positions between these keyframes.
- **Morph Targets (Blend Shapes)**: Define multiple sets of vertex positions and interpolate between them based on weights.

Keyframe Interpolation

- Vertices are defined at specific keyframes.
- Intermediate positions are calculated by interpolating between these keyframes.
- Commonly used for simple animations like doors opening or environmental effects.

MORPH TARGETS (BLEND SHAPES)

- Multiple versions of a mesh (targets) are created.
- Each target represents a different pose or shape.
- The final shape is a weighted blend of these targets.
- Widely used for facial animations to achieve detailed expressions.

Advantages of Vertex Animation

- Allows for detailed and complex deformations.
- Simple to implement and understand.
- No need for complex rigging or skeletal structures.

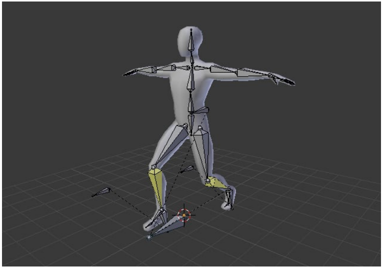
Disadvantages of Vertex Animation

- Can be memory intensive due to storing multiple vertex positions.
- Less flexible for character animation compared to skeletal animation.
- Interpolation artifacts can occur if not handled properly.

Skinning

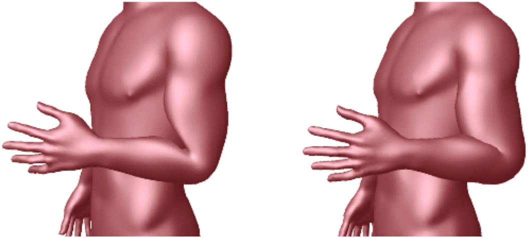
WHAT IS SKINNING?

- Skinning is a method used for character animation where a mesh (skin) is deformed based on the movement of an underlying skeleton (bones).
- Essential for creating realistic character movements.
- Allows for complex deformations driven by skeletal structures.



Linear Blend Skinning (LBS)

- Also known as smooth skinning.
- Each vertex is influenced by multiple bones.
- The final position is a weighted average of these influences.
- Simple and efficient but can cause artifacts like collapsing joints.



Dual Quaternion Skinning (DQS)

- An advanced technique to avoid artifacts of LBS.
- Uses dual quaternions (rotation and translation) for blending rotations, preserving volume.
- Provides smoother and more realistic deformations.
- Computationally more expensive but reduces issues like joint collapsing.

Rigid Skinning

- Simplest form of skinning.
- Each vertex is influenced by only one bone.
- Used for hard surfaces where smooth deformations are not required.

Skinning Matrices

- Bone transformations are represented as matrices.
- Vertices are transformed by these matrices based on bone weights.
- Ensures that skin follows the movement of bones accurately.

Advantages of Skinning Techniques

- Enables complex and realistic character animations.
- Efficient for real-time applications with proper optimization.
- Flexibility in animating both rigid and soft body characters.

Challenges in Skinning Techniques

- Requires careful weight painting to avoid deformation artifacts.
- Computationally intensive, especially for high-poly models.
- Complex rigging setup needed for detailed animations.

Physics-based Animation

What is Physics-Based Animation?

- Physics-based animation uses physical laws to simulate realistic movements and interactions in real-time.
- Adds realism to animations by mimicking real-world physics.
- Commonly used for particles, rigid bodies, fluids, cloth, and hair.

Particle Systems

- Simulate phenomena like fire, smoke, and explosions.
- Each particle represents a small part of the effect.
- Behavior governed by forces such as gravity, wind, and collision.
- Efficiently handled on the GPU for real-time performance.

Rigid Body Dynamics

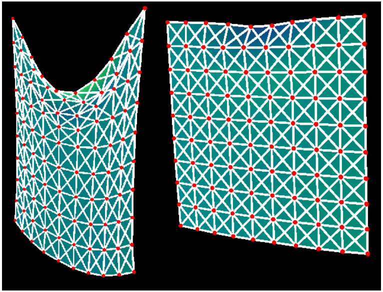
- Simulate the motion of solid objects.
- Objects can move, rotate, and collide with each other.
- Governed by Newton's laws of motion.
- Used for simulating objects like bouncing balls, falling debris, etc.

Fluid Simulations

- Create realistic water, liquid, and other fluid animations.
- Techniques include SPH (Smoothed Particle Hydrodynamics) and grid-based methods.
- Computationally intensive but can be optimized for real-time using the GPU.

Cloth Simulation

- Simulate the behavior of fabric as it moves and interacts with objects.
- Techniques include mass-spring systems and finite element methods (FEM).
- Used for realistic clothing, curtains, and other fabric materials.



Hair Simulation

- Simulate individual strands or clumps of hair.
- Techniques include particle-based methods and volumetric approaches.
- Ensures realistic movement and interactions with wind, gravity, and collisions.

Animation Blending

Why Use Animation Blending?

- Ensures smooth transitions between animations, enhancing realism.
- Prevents abrupt changes in movement that can break immersion.
- Allows for dynamic and responsive character behaviors.

Types of Animation Blending

- **Linear Blending**: Simple linear interpolation between two animations.
- **Non-Linear Blending**: More complex methods that consider the timing and trajectory differences between animations.
- **Additive Blending**: Adding small animation layers on top of a base animation for nuanced movements.

Linear Blending

- Interpolates linearly between keyframes of two animations.
- Simple and efficient.
- Suitable for straightforward transitions, e.g., from walking to running.

Non-Linear Blending

- Takes into account the differences in animation timing and trajectories.
- Produces more natural transitions.
- Often used in complex character rigs where animations need to be seamlessly integrated.

ADDITIVE BLENDING

- Allows for adding small, independent motions to a base animation.
- Useful for applying subtle adjustments, like breathing or hand movements.
- Enables reusability of base animations with different variations.

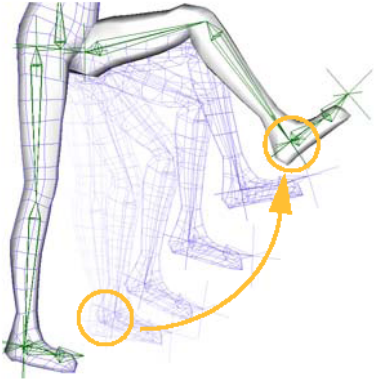
Challenges of Animation Blending

- Requires careful synchronization of animations to avoid visual artifacts.
- Performance can be impacted by complex blending operations.
- Managing multiple animation states and transitions can be complex.

Inverse Kinematics

WHAT IS INVERSE KINEMATICS (IK)?

- Inverse Kinematics (IK) is a technique used to calculate the necessary joint angles to achieve a desired position for a part of a character, such as a hand or foot.



Forward Kinematics vs Inverse Kinematics

- **Forward Kinematics (FK)**: Joint angles are given, and the position of each part is calculated.
- **Inverse Kinematics (IK)**: The desired position of an end-effector (e.g., hand, foot) is given, and the required joint angles are calculated.
- IK is often more intuitive for posing characters and creating interactions with the environment.

Applications of Inverse Kinematics

- **Character Animation**: Ensuring hands and feet reach target positions accurately.
- **Robotics**: Controlling robotic arms and legs to achieve precise movements.
- **Game Development**: Enabling characters to interact with objects and terrain dynamically.

IK Algorithms

- **Analytical Methods**: Solve IK problems using mathematical equations, providing exact solutions for simple kinematic chains.
- **Iterative Methods**: Use numerical techniques to approximate solutions, suitable for more complex kinematic chains.
- **CCD (Cyclic Coordinate Descent)**: Iteratively adjusts each joint angle to reduce the distance to the target.
- **FABRIK (Forward And Backward Reaching Inverse Kinematics)**: Solves IK by repeatedly adjusting joint positions in forward and backward passes.