Character Animation

COS 426, Spring 2018
Princeton University
Computer Animation

- Describing how 3D objects (& cameras) move over time
Computer Animation

- Challenge is balancing between …
  - Animator control
  - Physical realism
Computer Animation

- Manipulation
  - Posing
  - Configuration control

- Interpolation
  - Keyframes
  - In-betweens

https://blenderartists.org/

focus.gscept.com
Character Animation Methods

- Modeling (manipulation)
  - Deformation
  - Blendshapes
  - Skeletons

- Interpolation
  - Key-framing
  - Kinematics
  - Motion Capture

https://blenderartists.org/
Character Animation Methods

- Modeling (manipulation)
  - Deformation
  - Blendshapes
  - Skeletons

- Interpolation
  - Key-framing
  - Kinematics
  - Motion Capture
Deformation

• How to change a character’s pose?
  • Every vertex directly
  • Intuitive computation

https://www.youtube.com/watch?v=oxkf_N-QCNI
Deformation

- A HUGE variety of methods
  - Laplacian mesh editing
  - ARAP
  - CAGE Base
  - Barycentric coordinates
  - Heat diffusion
  - Variational
  - …
Deformation

• A HUGE variety of methods
  • Laplacian mesh editing
  • ARAP
  • CAGE Base
  • Barycentric coordinates
  • Heat diffusion
  • Variational
  • …
Laplacian Mesh Editing

- Local detail representation – enables **detail preservation** through various modeling tasks
- Representation with **sparse** matrices
- Efficient **linear** surface reconstruction
Overall framework

1. Compute differential representation

   \[ \delta_i = L(v_i) = v_i - \frac{1}{d_i} \sum_{j \in N(i)} v_j \]

2. Pose modeling constraints

   \[ v_i' = u_i, \quad i \in C \]

3. Reconstruct the surface – in least-squares sense

   \[
   \begin{pmatrix}
   L \\
   L_c
   \end{pmatrix} V = \begin{pmatrix}
   \delta \\
   U
   \end{pmatrix}
   \]
Differential coordinates?

• In matrix form:

\[ L_{ij} = \begin{cases} -w_{ij} & i \neq j \\ \Sigma_{j \in 1_{\text{ring}_i}} w_{ij} & i = j \\ 0 & \text{else} \end{cases} \]

• They represent the **local** detail / local shape description
  • The direction approximates the normal
  • The size approximates the mean curvature
Adding constraints

• In matrix form:

\[ L_{ij} = \begin{cases} 
-w_{ij} & i \neq j \\
\sum_{j \in 1 \text{ring}_i} w_{ij} & i = j \\
0 & \text{else}
\end{cases} \]
Adding constraints

- In matrix form:

\[
L_{ij} = \begin{cases} 
-w_{ij} & i \neq j \\
\sum_{j \in \text{ring}_i} w_{ij} & i = j \\
0 & \text{else}
\end{cases}
\]
Laplacian Mesh Editing

A short editing session with the *Octopus*
Character Animation Methods

• Modeling (manipulation)
  • Deformation
  • Blendshapes
  • Skeletons

• Interpolation
  • Key-framing
  • Kinematics
  • Motion Capture
Blendshapes

- Blendshapes are an approximate semantic parameterization
- Linear blend of predefined poses
Blendshapes

https://www.youtube.com/watch?v=KPDfMpuK2fQ
Blendshapes

• Usually used for difficult to pose complex deformations
  • Such faces

• Given:
  • A mesh $M = (V, E)$ with $m$ vertices
  • $n$ configurations of the same mesh, $M_b = (V_b, E), b = 1 \ldots n$

• A new configuration is simply:
  • $M' = (\Sigma_{b=1\ldots n} w_b V_b, E)$

• Delta formulation:
  • $M' = (\Sigma_{b=1\ldots n} V_0 + w_b (V_b - V_0), E)$
  • A bit more convenient

• $M_0$ - the rest pose, $w_b$ blend weights
Blendshapes

https://www.youtube.com/watch?v=zvUfiKQI5jQ
Character Animation Methods

- Modeling (manipulation)
  - Deformation
  - Blendshapes
  - Skeletons

- Interpolation
  - Key-framing
  - Kinematics
  - Motion Capture

https://blenderartists.org/

focus.gscept.com
Articulated Figures

• Character poses described by set of rigid bodies connected by “joints”
Articulated Figures

- Well-suited for humanoid characters

Rose et al. `96
Example: Ice Skating

(Mao Chen, Zaijin Guan, Zhiyan Liu, Xiaohu Qie,
CS426, Fall98, Princeton University)
Articulated Figures

• Animation focuses on joint angles, or general transformations
Character Animation Methods

- Modeling (manipulation)
  - Deformation
  - Blendshapes
  - Skeletons

- Interpolation
  - Key-framing
  - Kinematics
  - Motion Capture

https://blenderartists.org/
Forward Kinematics

• Describe motion of articulated character

\[ \begin{align*}
X &= (x,y) \\
\Theta_1, \Theta_2
\end{align*} \]
Forward Kinematics

- Animator specifies joint angles: $\Theta_1$ and $\Theta_2$
- Computer finds positions of end-effector: $X$

\[
X = (l_1 \cos \Theta_1 + l_2 \cos(\Theta_1 + \Theta_2), l_1 \sin \Theta_1 + l_2 \sin(\Theta_1 + \Theta_2))
\]
Forward Kinematics

- Joint motions specified e.g. by spline curves
Example: Walk Cycle

• Articulated figure:

- Hip
- Upper leg
- Knee
- Lower leg
- Ankle
- Foot

- Upper leg (hip rot)
  - Hip rotate
  - Lower leg (knee rot)
  - Hip rotate + knee rot
- Foot (ankle rot)
Example: Walk Cycle

- Hip joint orientation:

![Graph showing hip joint orientation with keyframes labeled 1 to 5.](image-url)
Example: Walk Cycle

• Knee joint orientation:

![Graph showing knee joint orientation with phases labeled 1, 2, 2B, 3, 3B, 4, 5.](image)
Example: Walk Cycle

• Ankle joint orientation:
Example: walk cycle

https://www.youtube.com/watch?v=DuUWxUitJos
Inverse Kinematics

• What if animator knows position of “end-effector”?

\[ X = (x, y) \]

\[ \theta_1 \]

\[ \theta_2 \]

End-Effector

(0,0)
Inverse Kinematics

• Animator specifies end-effector positions: \( X \)
• Computer finds joint angles: \( \Theta_1 \) and \( \Theta_2 \):

\[
\begin{align*}
\Theta_2 &= \cos^{-1}\left( \frac{x^2 + x^2 - l_1^2 - l_2^2}{2l_1l_2} \right) \\
\Theta_1 &= \frac{-(l_2 \sin(\Theta_2))x + (l_1 + l_2 \cos(\Theta_2))y}{(l_2 \sin(\Theta_2))y + (l_1 + l_2 \cos(\Theta_2))x}
\end{align*}
\]
Inverse Kinematics

- End-effector positions can be specified by spline curves

![Diagram of a two-link robot arm with joint angles \( \theta_1 \) and \( \theta_2 \), and end-effector position \( X = (x, y) \).]
Inverse Kinematics

- Problem for more complex structures
  - System of equations is usually under-constrained
  - Multiple solutions

\[ X = (x, y) \]

Three unknowns: \( \Theta_1, \Theta_2, \Theta_3 \)
Two equations: \( x, y \)
Inverse Kinematics

- Solution for more complex structures:
  - Find best solution (e.g., minimize energy in motion)
  - Non-linear optimization

\[ X = (x, y) \]
Kinematics

• Advantages
  • Simple to implement
  • Complete animator control

• Disadvantages
  • Motions may not follow physical laws
  • Tedious for animator

Lasseter `87
Kinematics

- **Advantages**
  - Simple to implement
  - Complete animator control

- **Disadvantages**
  - Motions may not follow physical laws
  - Tedious for animator

Lasseter `87
Beyond Skeletons…

• Skinning
Kinematic Skeletons

• Hierarchy of transformations (“bones”)
  • Changes to parent affect all descendent bones

• So far: bones affect objects in scene or parts of a mesh
  • Equivalently, each point on a mesh acted upon by one bone
  • Leads to discontinuities when parts of mesh animated

• Extension: each point on a mesh acted upon by more than one bone
Linear Blend Skinning

- Each vertex of skin potentially influenced by all bones
  - Normalized weight vector $w^{(v)}$ gives influence of each bone transform
  - When bones move, influenced vertices also move

- Computing a transformation $T_v$ for a skinned vertex
  - For each bone
    - Compute global bone transformation $T_b$ from transformation hierarchy
  - For each vertex
    - Take a linear combination of bone transforms
    - Apply transformation to vertex in original pose

$$T_v = \sum_{b \in B} w^{(v)}_b T_b$$

- Equivalently, transformed vertex position is weighted combination of positions transformed by bones

$$v_{\text{transformed}} = \sum_{b \in B} w^{(v)}_b (T_b v)$$
Assigning Weights: “Rigging”

• Painted by hand
• Automatic: function of relative distances to nearest bones
  • Smoothness of skinned surface depends on smoothness of weights!
Assigning Weights: “Rigging”

- Painted by hand
- Automatic: function of relative distances to nearest bones
  - Smoothness of skinned surface depends on smoothness of weights!
  - Other problems with extreme deformations
    - Many solutions
Assigning Weights: “Rigging”

• Painted by hand
• Automatic: function of relative distances to nearest bones
  • Smoothness of skinned surface depends on smoothness of weights!
  • Other problems with extreme deformations

Character Animation Methods

- Modeling (manipulation)
  - Deformation
  - Blendshapes
  - Skeletons

- Interpolation
  - Key-framing
  - Kinematics
  - Motion Capture

https://blenderartists.org/

focus.gscept.com
Keyframe Animation

- Define character poses at specific time steps called “keyframes”
Keyframe Animation

- Interpolate variables describing keyframes to determine poses for character in between

Lasseter '87
Keyframe Animation

• Inbetweening:
  • Linear interpolation - usually not enough continuity

H&B Figure 16.16
Keyframe Animation

- Inbetweening:
  - Spline interpolation - maybe good enough

H&B Figure 16.11
Example: Ball Boy

“Ballboy”

Fujito, Milliron, Ngan, & Sanocki
Princeton University
Character Animation Methods

- Modeling (manipulation)
  - Deformation
  - Blendshapes
  - Skeletons

- Interpolation
  - Key-framing
  - Kinematics
  - Motion Capture

https://blenderartists.org/
focus.gscept.com
Motion Capture

- Measure motion of real characters and then simply “play it back” with kinematics

Captured Motion
Motion Capture

- Measure human motion
- Play back with kinematics

https://www.youtube.com/watch?v=MVvDw15-3e8
Motion Capture

- Could be applied on different parameters
  - Skeleton Transformations
  - Direct mesh deformation

- Advantage:
  - Physical realism

- Challenge:
  - Animator control
Summary

• Kinematics
  ◦ Animator specifies poses (joint angles or positions) at keyframes and computer determines motion by kinematics and interpolation

• Dynamics
  ◦ Animator specifies physical attributes, constraints, and starting conditions and computer determines motion by physical simulation

• Motion capture
  ◦ Compute captures motion of real character and provides tools for animator to edit it