Character Animation

COS 426, Spring 2017
Princeton University

Amit H. Bermano
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
  • Animation
  • In-betweening
Character Animation Methods

• Modeling (manipulation)
  • Deformation
  • Blendshape rigging
  • Skeleton+Envelope rigging

• Interpolation
  • Key-framing
    • Kinematics
  • Motion Capture
  • Energy minimization
    • Physical simulation
  • Procedural

https://blenderartists.org/
Character Animation Methods

• Modeling (manipulation)
  • Deformation
  • Blendshape rigging
  • Skeleton+Envelope rigging

• Interpolation
  • Key-framing
    • Kinematics
  • Motion Capture
  • Energy minimization
    • Physical simulation
  • Procedural

https://blenderartists.org/

focus.gscept.com
Deformation

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

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

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

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 matricial form:

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

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

• In matricial form:

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

- In matricial form:

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

\[
\begin{bmatrix}
4 & -1 & -1 & -1 & -1 \\
-1 & 3 & -1 & -1 \\
-1 & -1 & 5 & -1 & -1 \\
-1 & -1 & 4 & -1 & -1 \\
-1 & -1 & 3 & -1 \\
-1 & -1 & -1 & 3 & -1 \\
-1 & -1 & -1 & -1 & 4 \\
1 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
x_7 \\
x_8 \\
x_9 \\
x_{10}
\end{bmatrix}
\begin{bmatrix}
\delta_1 \\
\delta_2 \\
\delta_3 \\
\delta_4 \\
\delta_5 \\
\delta_6 \\
\delta_7 \\
\delta_8 \\
\delta_9 \\
\delta_{10}
\end{bmatrix}
= 
\begin{bmatrix}
4 & -1 & -1 & -1 & -1 \\
-1 & 3 & -1 & -1 \\
-1 & -1 & 5 & -1 & -1 \\
-1 & -1 & 4 & -1 & -1 \\
-1 & -1 & -1 & 3 & -1 \\
-1 & -1 & -1 & -1 & 4 \\
1 & 1
\end{bmatrix}
\begin{bmatrix}
\delta_1 \\
\delta_2 \\
\delta_3 \\
\delta_4 \\
\delta_5 \\
\delta_6 \\
\delta_7 \\
\delta_8 \\
\delta_9 \\
\delta_{10}
\end{bmatrix}
= 
\begin{bmatrix}
\delta_1 \\
\delta_2 \\
\delta_3 \\
\delta_4 \\
\delta_5 \\
\delta_6 \\
\delta_7 \\
\delta_8 \\
\delta_9 \\
\delta_{10}
\end{bmatrix}
\begin{bmatrix}
P_x \\
Q_x \\
P_y \\
Q_y
\end{bmatrix}
\]
Laplacian Mesh Editing

A short editing session with the Octopus
Character Animation Methods

- Modeling (manipulation)
  - Deformation
  - Blendshape rigging
  - Skeleton+Envelope rigging

- Interpolation
  - Key-framing
    - Kinematics
  - Motion Capture
  - Energy minimization
    - Physical simulation
  - Procedural

https://blenderartists.org/

focus.gscept.com
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:
  - $M' = (\sum_{b=1}^n w_b V_b, E)$
- Delta formulation:
  - $M' = (\sum_{b=1}^n V_0 + w_b (V_b - V_0), E)$
  - A bit more convenient
- $M_0$ - the rest pose, $w_b$ blend weights
Blendshapes

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

• Modeling (manipulation)
  • Deformation
  • Blendshape rigging
  • Skeleton+Envelope rigging

• Interpolation
  • Key-framing
    • Kinematics
  • Motion Capture
  • Energy minimization
    • Physical simulation
  • Procedural

https://blenderartists.org/
Articulated Figures

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

Scene Graph

Angel Figures 8.8 & 8.9
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
Forward Kinematics

- Describe motion of articulated character

\[ X = (x, y) \]

“End-Effector”
Forward Kinematics

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

$$X = (l_1 \cos \Theta_1 + l_2 \cos(\Theta_1 + \Theta_2), l_1 \sin \Theta_1 + l_2 \sin(\Theta_1 + \Theta_2))$$
Example: Walk Cycle

• Articulated figure:

```
  Hip
  |
  v
Upper leg
  |
  v
  Knee
  |
  v
Lower leg
  |
  v
  Ankle
  |
  v
  Foot

  Upper leg (hip rot)
  |
  v
  Hip rotate
  |
  v
  Lower leg (knee rot)
  |
  v
  Hip rotate + knee rot
  |
  v
  Foot (ankle rot)
```

Watt & Watt
Example: Walk Cycle

- Hip joint orientation:
Example: Walk Cycle

• Knee joint orientation:

Watt & Watt
Example: Walk Cycle

- Ankle joint orientation:
Example: walk cycle

Lague: www.youtube.com/watch?v=DuUWxUitJos
Character Animation Methods

• Modeling (manipulation)
  • Deformation
  • Blendshape rigging
  • Skeleton+Envelope rigging

• Interpolation
  • Key-framing
    • Kinematics
  • Motion Capture
  • Energy minimization
    • Physical simulation
  • Procedural

https://blenderartists.org/

focus.gscept.com
Beyond Skeletons…

• Skinning
Kinematic Skeletons

• Hierarchy of transformations ("bones")
  • Changes to parent affect all descendant 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

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

![Comparison of linear (left) and dual quaternion (right) blending. Dual quaternions preserve rigidity of input transformations and therefore avoid skin collapsing artifacts.](image)
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
  - Blendshape rigging
  - Skeleton+Envelope rigging

- **Interpolation**
  - Key-framing
    - Kinematics
  - Motion Capture
  - Energy minimization
    - Physical simulation
  - Procedural
Keyframe Animation

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

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

• Inbetweening:
  • Linear interpolation - usually not enough continuity
Keyframe Animation

- Inbetweening:
  - Spline interpolation - maybe good enough

H&B Figure 16.11
Temporal Enhancement

Start frame

End frame
Temporal Enhancement

Displacement

Time

Database

Interpolation
Results – Hand Animated Rig

Database

Linear

Our Enhancement
Example: Ball Boy

“Ballboy”

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

• Modeling (manipulation)
  • Deformation
  • Blendshape rigging
  • Skeleton+Envelope rigging

• Interpolation
  • Key-framing
    • Kinematics
  • Motion Capture
  • Energy minimization
    • Physical simulation
  • Procedural

https://blenderartists.org/
Inverse Kinematics

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

\[
\begin{align*}
\theta_1 & \\
\theta_2 & \\
X &= (x, y) \\
\end{align*}
\]
Inverse Kinematics

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

\[
\Theta_1 = \frac{-(l_2 \sin(\Theta_2))x + (l_1 + l_2 \cos(\Theta_2))y}{(l_2 \sin(\Theta_2))y + (l_1 + l_2 \cos(\Theta_2))x}
\]

\[
\Theta_2 = \cos^{-1}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}\right)
\]
Inverse Kinematics

• End-effector positions can be specified by spline curves
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
Character Animation Methods

- Modeling (manipulation)
  - Deformation
  - Blendshape rigging
  - Skeleton+Envelope rigging

- Interpolation
  - Key-framing
    - Kinematics
  - Motion Capture
  - Energy minimization
    - Physical simulation
  - Procedural

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

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

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

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

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

• Advantage:
  • Physical realism

• Challenge:
  • Animator control
Character Animation Methods

• Modeling (manipulation)
  • Deformation
  • Blendshape rigging
  • Skeleton+Envelope rigging

• Interpolation
  • Key-framing
    • Kinematics
  • Motion Capture
  • Energy minimization
    • Physical simulation
  • Procedural

https://blenderartists.org/

focus.gscept.com
Character Animation Methods

• Modeling (manipulation)
  • Deformation
  • Blendshape rigging
  • Skeleton+Envelope rigging

• Interpolation
  • Key-framing
    • Kinematics
  • Motion Capture
  • Energy minimization
    • Physical simulation

https://blenderartists.org/
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