Physically Based Motion Transformation
Digital Character Animation

- **Character** – animated object with a number of degrees of freedom (DOFs)
- **Motion** – set of functions $q(t)$ depicting how each DOF changes through time
The Animation Problem

Automatic generation of expressive/realistic motion that achieves a given set of tasks

- An open problem
- Realism vs. control tradeoff
Physically-based Methods

- Forward simulation [Baraff]
  - Highly realistic
  - Simulated character very hard to control

- Controllers [Raibert, Hodgins, Ngo, van de Pane]
  - Fast motion generation once controllers are computed
  - No set rules on controller generation
Spacetime Constraints

- Provide both realism and control
- Downside
  - Methods do not scale up
  - Sensitivity to the initial position
Captured Motion

- Sampled DOFs through time gathered from the real world
- Rich and realistic
- Hard to edit
Motion Warping

- Set poses which the warped motion should interpolate
- Set time constraints and solve for the minimum curve deviation. Interpolate the space constructed by a few sample motion capture sequences
High Level Control

- Get a limp walk by making one leg stiff
- Reduce gravity to get a “moon walk”
- Change the position and timing of foot placements
- Make a “quiet” run by reducing the floor impact forces
The New Approach

- Transform existing motion
- Spacetime constraints formulation
- Simplified character representation
- Get the best of both worlds:
  - Expressiveness of captured data
  - Controllability of the spacetime model
Outline

Complex Model

Original motion → Final motion

Motion Library

Fitting

Spacetime motion model

Δ

Reconstruction

Simplified Model

Transformed spacetime motion

Spacetime Editing

Original motion

Final motion
Outline

Original motion

Δ

Fitting

Spacetime motion model

Spacetime Editing

Transformed spacetime motion

Final motion

Reconstruction

Complex Model

Simplified Model

12
Model Fitting

Two phases:

- Simplify character kinematics
- Use input motion to construct a spacetime motion model
Simplified Kinematics

- Remove irrelevant DOFs
- Reduce *passive* body structure to mass points
- Exploit symmetric movement of limbs
Simplified Kinematics

Human Run

Human Jump
Motion Fitting

- **Handle** – a property that correlates the original and simplified model
- Must have enough handles to fully determine simplified model configuration
Handle Examples
Motion Synthesis As Constrained Optimization

- Body, muscle and force DOFs: $q(t)$
- Constraints:
  - Pose $C_p$
  - Mechanical $C_m$
  - Dynamics $C_d$
- Objective $E(q(t))$
Spacetime Model Fitting

- Biological data: mass distribution, muscles
- Use *handles* to create “best-guess” motion
- Specify constraints essential for given motion (e.g. foot placements)
- Use simple objective: smooth muscles

\[ E(q) = \ddot{q}^2 \]
Outline

Original motion

Fitting

Motion Library

Spacetime motion model

Spacetime Editing

Transformed spacetime motion

Final motion

Reconstruction

Simplified Model

Complex Model
Spacetime Editing

- Change pose and environment constraints
  - Foot placement and timing
  - Introduce a new obstacle
- Change the objective function
  - Minimize floor impact forces
  - Make dynamic balance more important
Spacetime Editing

- Change explicit character parameters
  - Short leg
  - Redistribute mass
  - Modify muscle characteristic
  - Gravity
Motion Reconstruction

Three different handle sets
- Original motion handles $h(q_o)$
- Spacetime fit handles $h(q_s)$
- Transformed spacetime handles $h(q_t)$

Compute final motion handles

$h(q_f) = h(q_o) + (h(q_t) - h(q_s))$
Minimum Displaced Mass Objective

$E_{dm}(q_o, q)$ evaluates total displaced mass when moving a character from pose $q_o$ to pose $q$

$$E_{dm} = \iiint_{i} \mu_i \left( p_i(q_o) - p_i(q) \right)^2 dx \, dy \, dz$$
For each time $t$ solve

\[
\begin{align*}
\text{minimize} & \quad E_{dm}(q_o, q_f) \\
\text{subject to} & \quad h(q_f) = h(q_o) + (h(q_t) - h(q_s))
\end{align*}
\]
Alternative Reconstruction Algorithm

For each time $t$ solve

$$\begin{align*}
\text{minimize} \quad & w_d E_d (q_o, q_f) + \\
\quad & w_h \left[ (h(q_f) - h(q_o)) - (h(q_t) - h(q_s)) \right]^2
\end{align*}$$
Example: Human Run

- Original model has 59 DOFs
- Simplified model has 19 DOFs
- Optimizations are done on one gait cycle
- Each optimization completes within 2 minutes
Biped

- Hinge Joint
- Ball Joint
Example: Human Broad Jump

- Original model has 59 DOFs
- Simplified model has 11 DOFs
- Entire upper body reduced to a mass point
- No joint angle DOFs
Hopper

Prismatic Joint
Future Work

- Optimal robots
- Extracting style
- Motion retargeting
- Building motion libraries
- Digital actors