On-line Puppetry: Issues and Solutions

Sung Yong Shin
Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology
http://cg.kaist.ac.kr

1. What is “On-line Puppetry”?
David Struman, Computer Puppetry, IEEE CG&A

Computer Puppetry

Manipulating the posture of the puppet directly
Displaying a sequence of postures in real time

Bringing an inanimate object (puppet) into life.

On-line (Computer) Puppetry

On-line: Future data are not available
- Inherent nature of puppetry

(Off-line) Motion Retargetting

M. Gleicher, SIGGRAPH ’98
"Off-line" is implicit in motion retargetting

On-line Motion Retargetting

Choi and Ko, Pacific Graphics ’99
(Off-line) Motion Retargetting

Retargetting:
  adapting the motion of an articulated figure to another figure
  with identical structure but different segment lengths

Spacetime Constraint Formulation [Gleicher 98]
  • Interactive problem specification
    • Characteristic features to preserve
    • Spatial constraints
    • Temporal constraints
  • Off-line: one large problem over the duration of motion
    ∴ look-ahead and -behind to avoid “jerkiness”

Clean input motion

On-line Puppetry: Issues

Automatic problem specification ★ Importance Analysis

On-line: A small problem at each frame
  • Look-ahead ★ Importance Analysis
  • Real-time performance ★ Real-time IK Solver

Noisy input motion ★ Kalman Filter

On-line Puppetry: Solution

Captured motion signal → Kalman filter → Importance analysis → IK solver → Root position estimation → Final motion
  Body posture computation
  Limb posture computation
2. Importance Analysis

\[ \mathbf{m}(t) = ([p_i(t), q_i(t)], [p_i(t), q_i(t)], \ldots, [p_i(t), q_i(t)]) \]

Which ones are important to preserve?
- The position of the root of the character
- The joint angles
- The positions of the end-effectors

If an end-effector interacts with an object in the environment, the position of the end-effector is important. Otherwise, the joint angles on its corresponding limb are important.

The distance from an end-effector to the object gives a clue to determine what to preserve!!

**Importance Computation**

\[
\bar{d}_i(t) = \frac{d_i(t) + \Delta d_i(t)}{D_i}
\]

\[
p_i(0) = 1, \quad p_i(1) = 0, \quad \bar{d}_i(0) = 0, \quad \bar{d}_i(1) = 0
\]

\[
p_i(\bar{d}_i) = \begin{cases} 
2\bar{d}_i^2 - 3\bar{d}_i^3 + 1, & \text{if } \bar{d}_i \leq 1 \\
0, & \text{otherwise}
\end{cases}
\]

\[
w_i = \max_j(\bar{d}_j)
\]
3. Real-time Inverse Kinematics Solver

Three step approach
- Root Position Estimation
- Body Posture Computation
- Limb Posture Computation

3.1 Root Position Estimation

\[ d = \frac{\sum w_i d_i}{\sum w_i} \]

(Offsetting Adjustment)

Reachable Ranges

End-effector position
Shoulder position
Root position
3.2 Body Posture Computation

Needed if the root position fails to make all end-effectors reachable to their goal

Body posture:
- Root position
- Root orientation
- Orientations of body segments

\[(p_0, q_0, q_1, \ldots, q_n)\]

For a rigid torso, this is reduced to

\[(p_0, q_0, q_1)\]

Optimization Formulation

\[E = E_g + \alpha E_{\gamma}\]

\[E_g = \gamma \|p_s - p_g\|^2 + \sum \beta_j \|\ln(q_j)\|^2\]

\[E_{\gamma} = \sum E_i, \text{ where } E_{i} = \begin{cases} 0, & \text{if } \|p_i - p_i^f\| < r_i \\ \|p_i - p_i^f\|^2 - r_i^2, & \text{otherwise} \end{cases}\]
3.3 Limb Posture Computation

Posture blending

\[ q' = \text{slerp}(q, q^*, w) \]
\[ = (q \cdot q^{-1})^* \cdot q^* \]

Analytic Limb IK Solver

Elbow circle

Give the swivel angle \( \theta \),
the limb posture is analytically computed

\[ q' = r(q, q^*, w) \]
\[ = (q \cdot q^{-1})^* \cdot q^* \]

Analytic Limb IK Solver

How to compute \( \theta \)

\[ \phi(\theta) = \min(\cos^{-1}(q^* \cdot q(\theta)), \cos^{-1}(-q^* \cdot q(\theta))) \]

where \( q(\theta) = \alpha \cdot q \),
\[ |q^* \cdot q(\theta)| \text{ is maximized} \]
\[ \| \frac{\theta}{2} + \alpha \| \]
\[ \therefore \theta \text{ can be compute analytically} \]
4. Noise Filtering

Kalman filter: least square estimator for a linear system
However, the orientation space is not linear.
∴ Locally linearize the orientation space !!!

Incremental orientation
• Euler angles [Welch et al 97]
• Rotation vector
  \[ q_j(t) = e^{q_j(t-1)} \]
  \[ v_j = \ln(q_j(t)q_j^{-1}(t-\Delta t)) \]
• The state for the \( j \)'s sensor: \( z_j = (p_j, \dot{p}_j, v_j, \dot{v}_j) \)

5. Discussion

• The importance of an end-effector guides what to preserve.
• It also gives how urgent the interaction of an end-effector with the environment.
  ∴ Automatic constraint detection
• It has a limited look-ahead capability.
• The specialized IK solver maps the posture in real-time.
  However, ...
• The environment needs to be not too complex for real-time performance.
• The IK solver is specialized for human-like figures.

6. Conclusion and Future Work

Practical solution
  KBS "Pang Pang", "Aliang"

Self-interaction
Non-human-like characters