Motion Synthesis and Editing

Yisheng Chen

Overview

- Data driven motion synthesis
  - automatically generate motion from a motion capture database, offline or interactive
  - User inputs
  - Large, high-dimensional database
- Motion editing
  - Change how things move: good motion → different good motion
  - Complementary to motion synthesis

Papers

- Interpolated Motion Graphs
- Constraint-Based Motion Optimization
- Graph Optimal Control
- Guided Time Warping

Overview

- Generate animation for novice animators
  - Children, trainers, game players
  - Small amount of user input
    - paths, constraints
- Key ideas
  - Interpolated motion graphs
  - Global optimal solution
  - Compression and heuristic search

Discrete Optimization Problem

- Unknowns
  - poses of the character over time
  - $P_1(t)$, $P_2(t)$, $w(t)$

$$P_{new}(t) = P_1(t)w + P_2(t)(1-w)$$

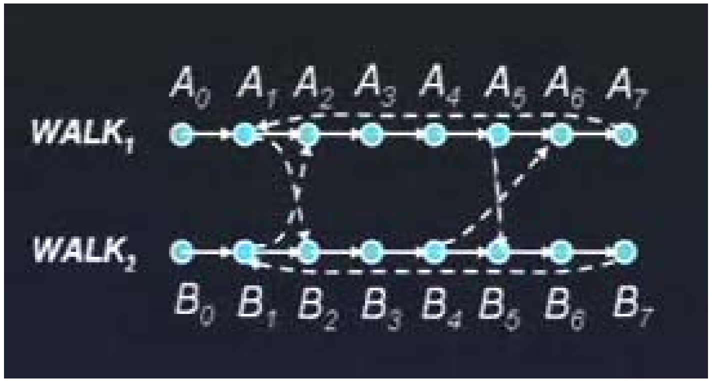
- Minimize
  - Efficiency, sum of squared torques
- Constraints
  - user, environmental and physics
- Search space is very large
  - 50 * T unknowns

Continuous constrained optimization

- Liu and Popovic 02, Fang and Pollard 03, Anderson and Pandy 99, Gleicher 97, Popovic and Witkin 99, Sulejmanpasic and Popovic 04, Lee and Shin 99, Safonova, Pollard and Hodgins 04 …
- Works well
  - When good initial guess is provided
  - For short single behavior motions
  - For simple characters
- Long, multi-behavior motion

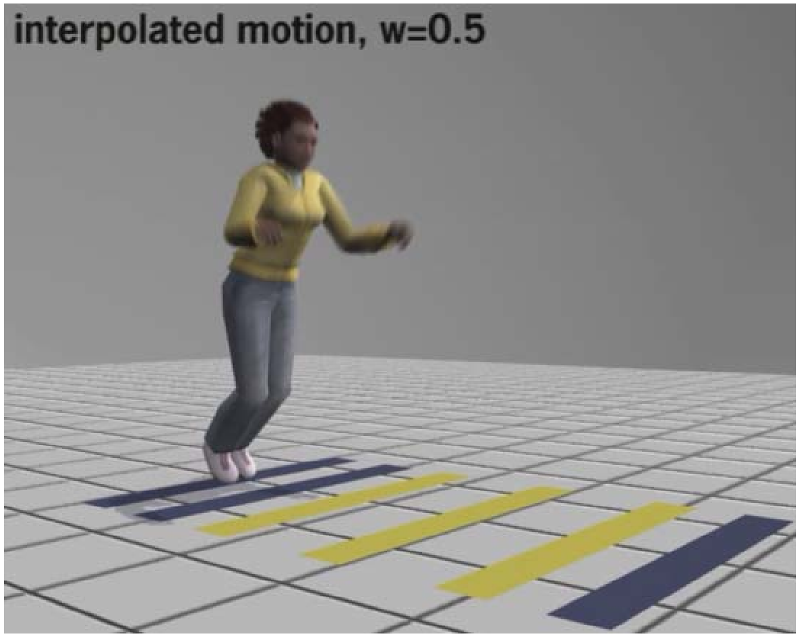
Motion Graphs

- Create long multi-behavior motions
- Captures natural transitions
- Cannot synthesize variations



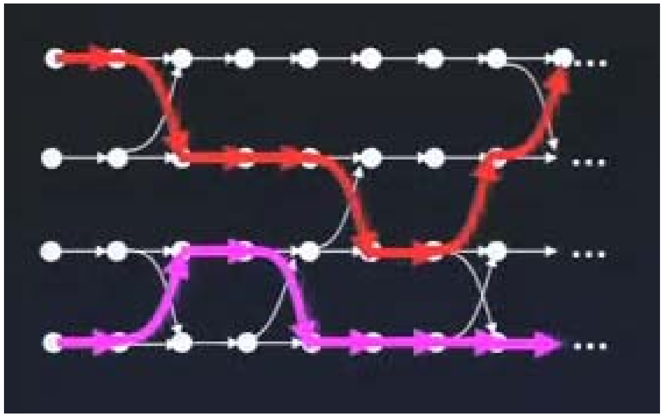
Motion Interpolation

- Resulting motion is natural
- Proved: closed to physically correct in many cases
- Sequences must be aligned and of a single behavior

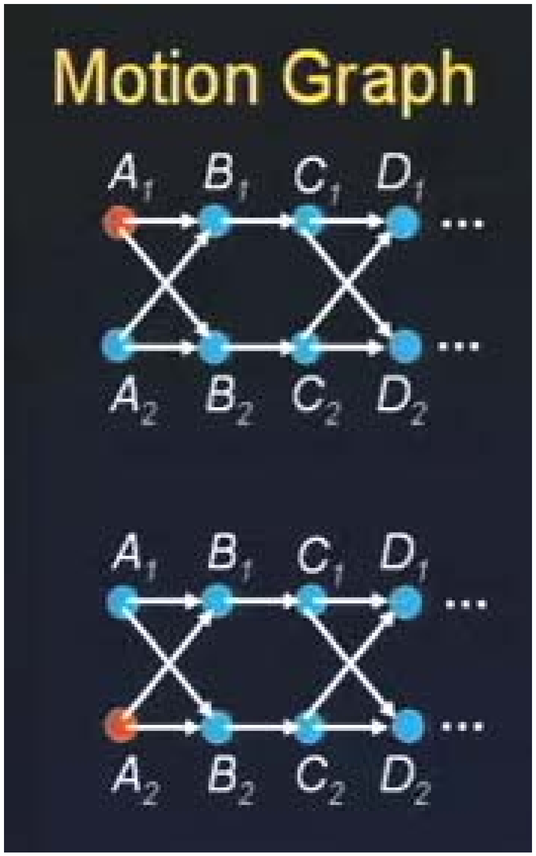


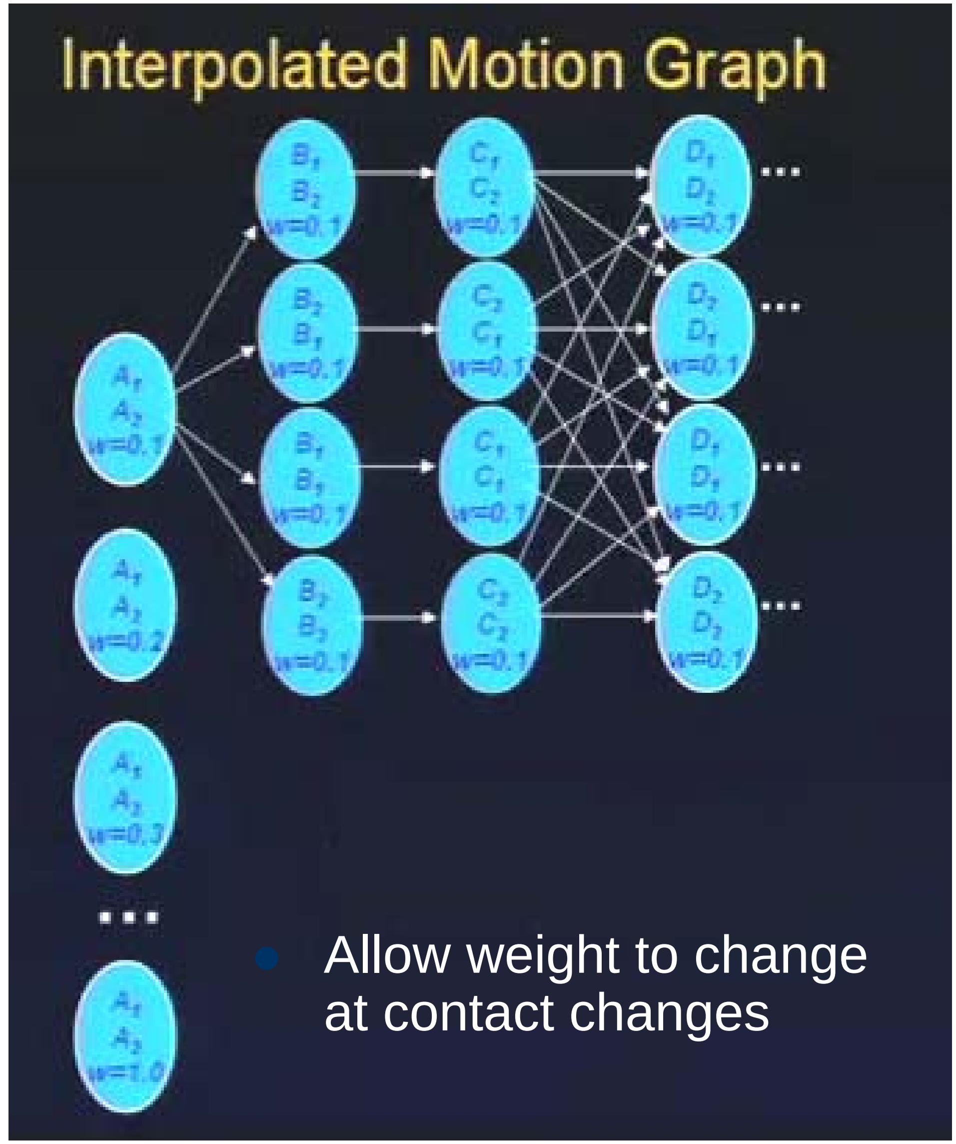
Interpolated motion graph

- Discrete compact representation of motion
  - Long, multi-behavior motions
  - Novel motion
  - No need to cut motions to segments
  - Retain naturalness



$$M'(t) = w(t)M_1(t) + (1 - w(t))M_2(t).$$


Motion Graph
A_1 B_1 C_1 D_1 ...
A_2 B_2 C_2 D_2 ...
A_1 B_1 C_1 D_1 ...
A_2 B_2 C_2 D_2 ...


Interpolated Motion Graph
- Allow weight to change at contact changes

Related work: Motion graphs and interpolation

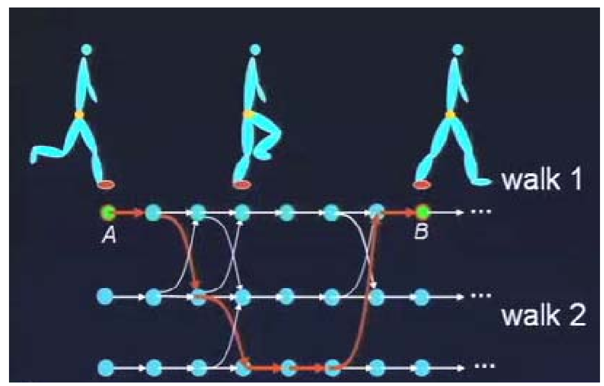
- Structured / unstructured graphs
- Interactive user control / synthesis based on sketch
  - Motion modeling for on-line locomotion synthesis [kwon and Shin]
  - Fat graphs: constructing an interactive character with continuous controls [Shin and Oh]
  - On-line locomotion generation based on motion blending [Park, Shin and Shin]
  - Parametric motion graphs [Heck and Gleicher]

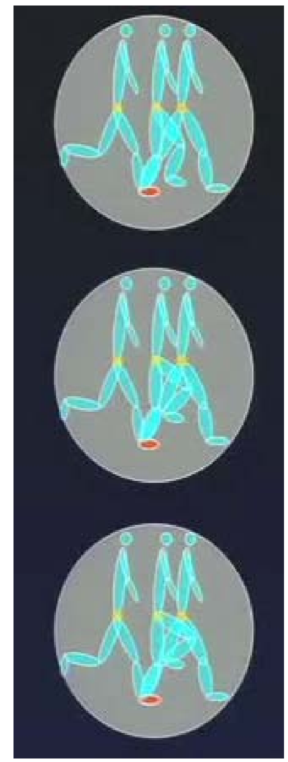
Searching motion graphs

- During search unroll into environment
  - (pose, position, orientation)
- Existing approaches for motion graphs
  - Use local search
  - Global sub-optimal search
- Interpolated motion graph is even more challenging to search
- Compress motion graph
  - Cull sub-optimal paths
  - Cull redundant data
  - Functionally equivalent but much smaller graph

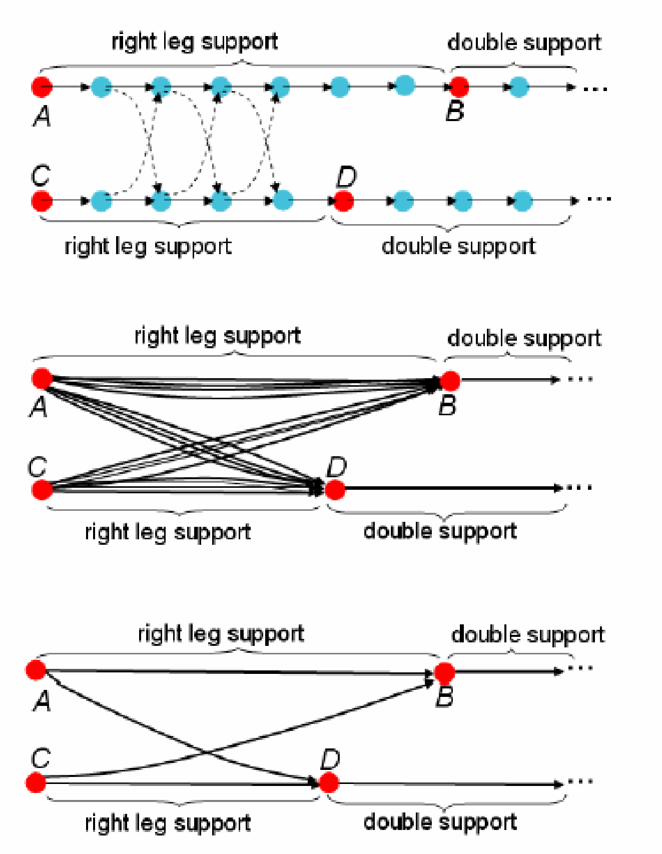
Cull sub-optimal paths

- There are many paths from A to B
  - Consider only paths that share a contact
  - Contact change (different set of contacts with environment)
  - Thousands of such paths, most are very similar
  - All paths end at the same root position/orientation
  - Only leave one path that minimizes the objective function





Cull redundant states



what are lost by culling?

- Variations
  - Suboptimal, would not have been selected anyway
- Constraints in the middle of contact phase
  - Often fall at contact phase
- If necessary, revert to the original graph

Search Graph

- Search method
  - A*
- Derived information heuristic function that guides the search
  - Location, pose
- Key to being able to find optimal solutions
  - Compression + information heuristic function

Summary

- Interpolated motion graph
  - Long, multi-behavior motions
  - Can compute novel motions
  - Can search for nearly-optimal solutions
    - Compression
    - Heuristic function
  - Search techniques also apply to original motion graphs
- Future work
  - Larger database
  - Longer motions
  - Leave more variations in the graphs
  - Blend more that two paths

Papers

- Interpolated Motion Graphs
- Constraint-Based Motion Optimization
- Graph Optimal Control
- Guided Time Warping

Human animation from user constraints

- Goal: generate natural human motion from spatial-temporal constraints
  - Key frames
  - Key trajectories
- Challenges
  - Human motions are high-dimensional while constraints are not
  - People are experts on natural motion

Problem Formulation

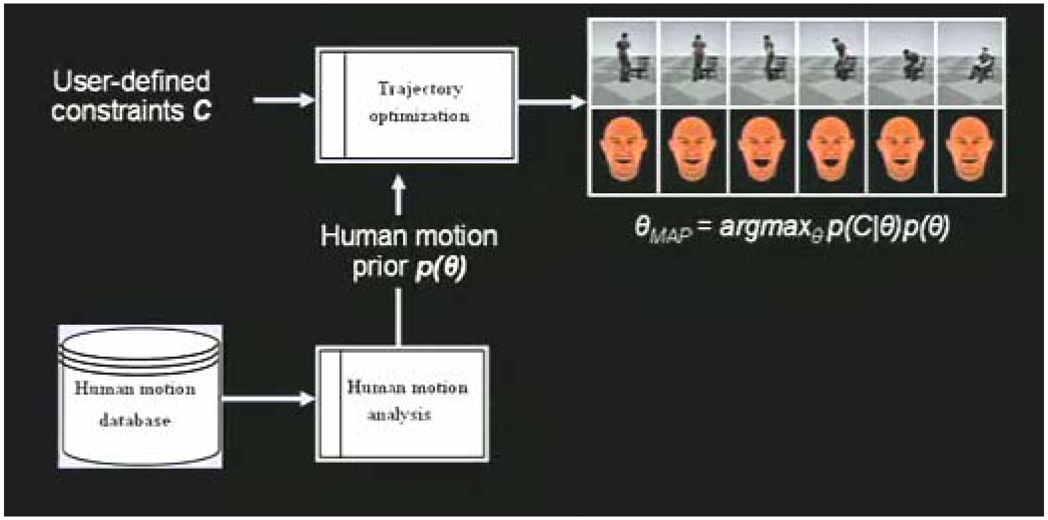
- Find the most likely human motion θ_{MAP} from user constraints C:

$$\theta_{MAP} = argmax_{\theta}\, p(\theta|C) \qquad \text{(Maximum a posterior)}$$

$$= argmax_{\theta}\, \underbrace{p(C|\theta)}_{\text{How well motion matches Constraints?}} \times \underbrace{p(\theta)}_{\text{How natural motion is generated?}} \qquad \text{(Bayes rule)}$$

Algorithm overview

- Generate natural human motion from user-defined constraints



Related work

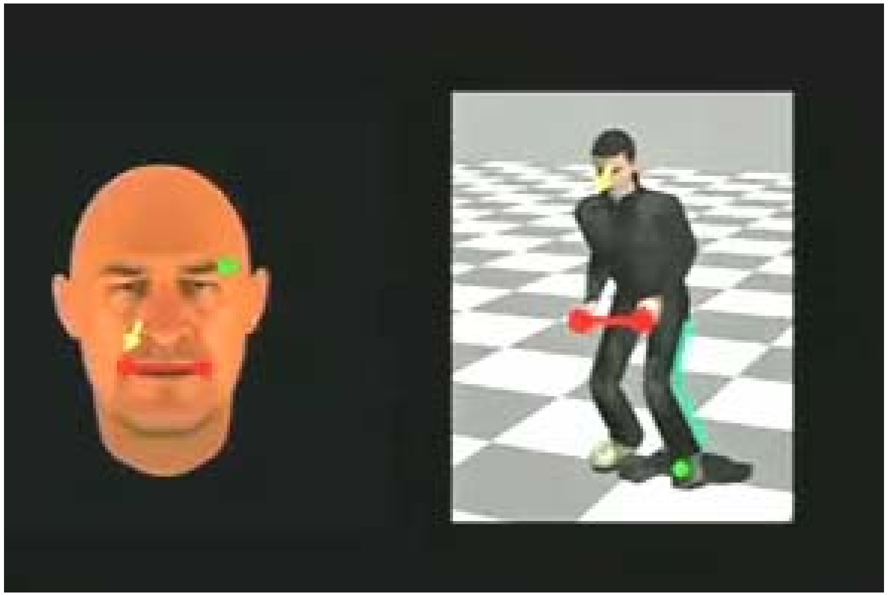
- Physically based trajectory optimization
- Animation from motion capture data
  - Reordering motion clip
  - Learning model from human motion
  - Interpolating motions

Human motion data

- A prerecorded large and heterogeneous human motion database
- Full-body movement (1 hour) and facial movement (10 minutes)

User defined constraints

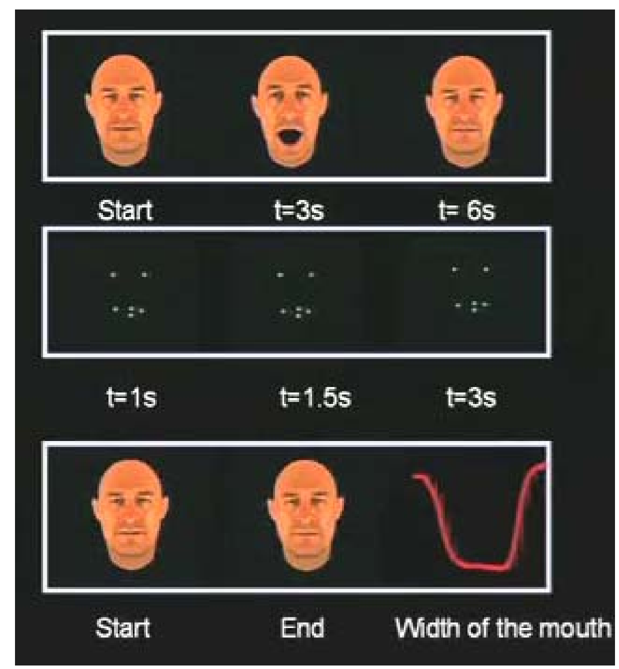
- Any kinematic constraints throughout the motion
  - Position
  - Orientation
  - Distance
  - Joint and angle



Constraint examples

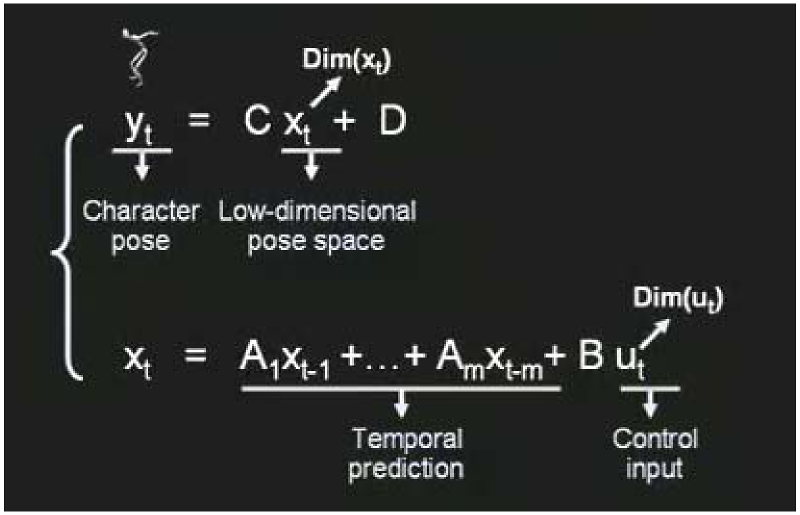
- Body animation

- Face animation



Statistical dynamic model

- Statistical dynamic model
  - PCA
  - Linear time-invariant system
- Model complexity
- Machine learning
  - Dynamic model matrices
    - $A_1, \ldots, A_m$, B, C, D
  - Dynamic system order
    - m, dimensionality of x_t and u_t



Control input

$$\hat{A}_1, ..., \hat{A}_m \quad = \quad \arg\min_{A_1,...,A_m} \sum_n \|\mathbf{u}_n\|^2$$

$$\hat{A}_1, ..., \hat{A}_m = \arg\min_{A_1,...,A_m} \sum_{n=m+1}^{N} \|\mathbf{x}_n - \sum_{i=1}^{m} A_i \mathbf{x}_{n-i}\|^2$$

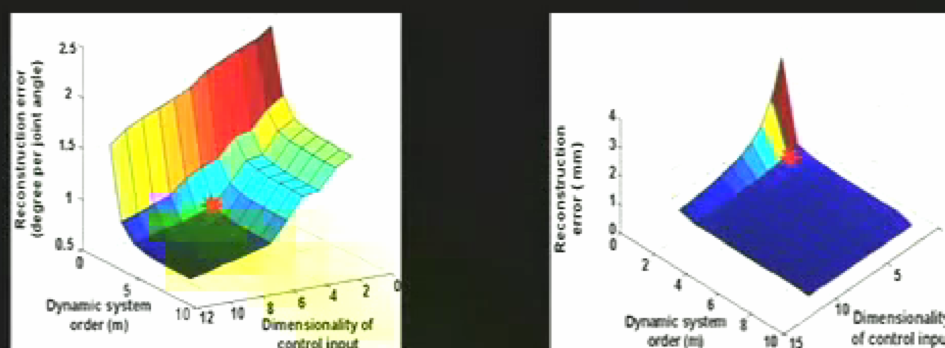
$$\hat{\mathbf{z}}_n = \mathbf{x}_n - \sum_{i=1}^{m} \hat{A}_i \mathbf{x}_{n-i}, \quad n = m+1, ..., N$$

$$\underbrace{\left(\hat{\mathbf{z}}_{m+1} \mid ... \mid \hat{\mathbf{z}}_N \right)}_{Z} \quad = \quad B \cdot \underbrace{\left(\mathbf{u}_{m+1} \mid ... \mid \mathbf{u}_N \right)}_{U}$$

Reconstruction error

- Statistical dynamic model

$$\begin{cases} Y_t = C X_t + D \\ X_t = A_1 X_{t-1} + \ldots + A_m X_{t-m} + B\, u_t \end{cases}$$



Full-body walking data

$m = 3$, $\dim(u_t) = 4$, error = 0.7 deg

Facial data

$m = 2$, $\dim(u_t) = 1$, error = 0.1 mm

Forward simulation

- Full body movement

$$\begin{cases} y_t = C\,x_t + D \\ x_t = A_1 x_{t-1} + A_2 x_{t-2} + A_3 x_{t-3} + B\,u_t \end{cases}$$

$\mathrm{Dim}(u_t) = 4$

- Facial movement

$$\begin{cases} y_t = C\,x_t + D \\ x_t = A_1 x_{t-1} + A_2 x_{t-2} + B\,u_t \end{cases}$$

$\mathrm{Dim}(u_t) = 1$

Human motion prior

- The motion prior of generated motion sequence
- Assume u_t is independent each other
- x_t depends on the x_{t-1}, …, x_{t-m}, and u_t

$$
\begin{aligned}
p(\theta) &= p(x_1,\ldots,x_T,u_1,\ldots,u_T) \\
&\cong \underbrace{\prod p(u_t)}_{\text{Control prior}} \times \underbrace{\prod p(x_t \mid x_{t-1},\ldots,x_{t-m},u_t)}_{\text{Dynamics prior}} \qquad \text{(Chain rule)}
\end{aligned}
$$

Motion Optimization

- Find the most likely motion θ_{MAP} from user constraints C:
  - Trajectory optimization
  - Cubic B-splines for x and u
  - Sequential quadratic programming
  - Random initial guess [0…1]

$$\theta_{MAP} = \mathrm{argmin}_{\theta} \underbrace{-\ln p(C|\theta)}_{\text{Constraint term}} - \underbrace{\sum \ln p(u_t)}_{\text{Prior for control input}} - \underbrace{\sum \ln p(x_t|x_{t-1}, \ldots, x_{t-m}, u_t)}_{\text{Prior from statistical dynamic model}}$$

Objective functions

$$
\begin{aligned}
E_{constraints} &= -\ln p(E|H) \\
&\sim \sum_{j=1}^{J} \beta \|\mathbf{e}_j - \mathbf{f}_j(\mathbf{y}_1, ..., \mathbf{y}_T)\|^2 \\
&\sim \sum_{j=1}^{J} \beta \|\mathbf{e}_j - \mathbf{f}_j(C\mathbf{x}_1 + D, ..., C\mathbf{x}_T + D)\|^2
\end{aligned}
$$

$$
\begin{aligned}
E_{prior}^{dynamic} &= -\ln \prod_{t=m+1}^{T} p(\mathbf{x}_t | \mathbf{x}_{t-1:t-m}, \mathbf{u}_t) \\
&\sim -\alpha \sum_{t=m+1}^{T} \|\mathbf{x}_t - \sum_{i=1}^{m} A_i \mathbf{x}_{t-i} - B\mathbf{u}_t\|^2
\end{aligned}
$$

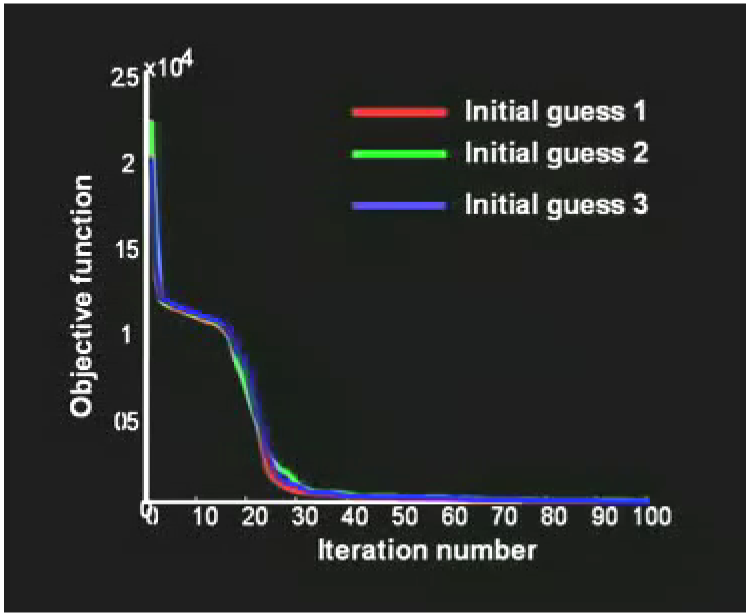
$$
\begin{aligned}
E_{prior}^{control} &= -\ln p(\mathbf{x}_{1:m}, \mathbf{u}_{m+1:T}) \\
&= -\sum_{t=1}^{m} \ln p(\mathbf{x}_t) - \sum_{t=m+1}^{T} \ln p(\mathbf{u}_t)
\end{aligned}
$$

$$p(\mathbf{u}_t) = \Sigma_{k=1}^{K} \pi_k N(\mathbf{u}_t; \phi_k, \Lambda_k)$$

$$\arg\min_{\mathbf{x},\mathbf{u}} \quad E_{constraint} + E_{prior}^{dynamic} + E_{prior}^{control}$$

Optimization

- The evolution of the objective function values with three different initial guesses

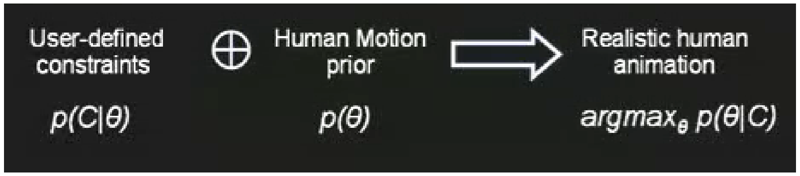


Summary

- An alternative for physically based optimization
- Does not need physical models
- Low-dimensional model and faster convergence
- Can generate slow even stylized motion
- Requires appropriate mocap data
- Cannot specify dynamic constraints (e.g., mass)
- Intuitive interfaces for specifying spatial-temporal constraints?

Conclusions

- Trajectory optimization with statistical models



- A new compact spatial-temporal representation for human data

$$Y_t = C X_t + D$$

$$X_t = A_1 X_{t-1} + \ldots + A_m X_{t-m} + B u_t$$

Papers

- Interpolated Motion Graphs
- Constraint-Based Motion Optimization
- Graph Optimal Control
- Guided Time Warping

Near-optimal Character Animation with Continuous Control

- Generate real-time character animation with interactive control
- Challenges
  - Planning
    - Walking and turning
    - Obstacle avoidance
  - Optimality
  - Real-time
  - Dimensionality

Goals

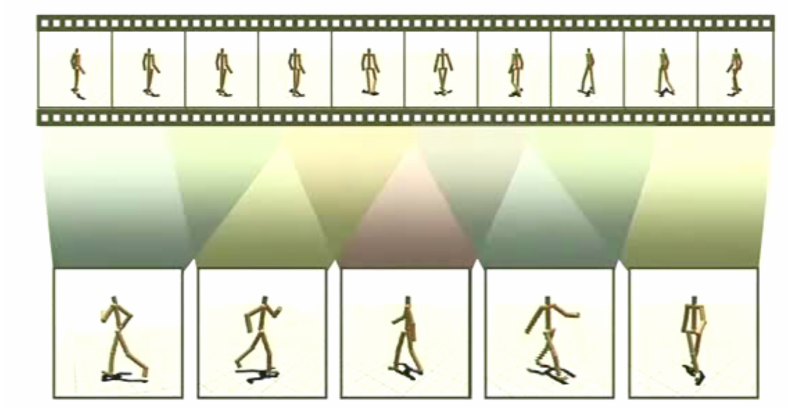
- Many dimensions
- Several tasks
- Simple
- Responsive, real-time
- Near-optimal planning

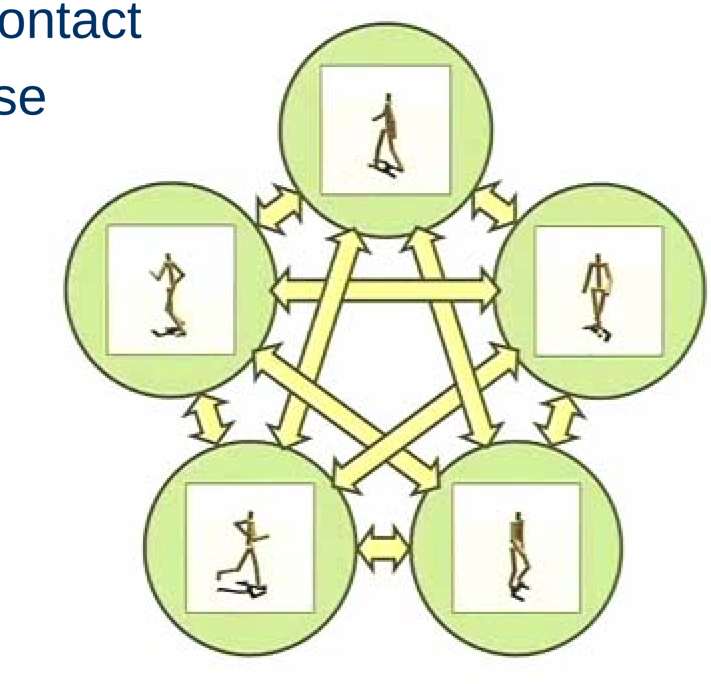
Related work

- Off-line planning
  - Motion synthesis from annotations [Arikan et al, SIGGRAPH 2003]
  - Interpolated motion graphs [Safonova and Hodgins, SIGGRAPH 2007]
- Real time planning
  - Precomputed search trees [Lau et al, SCA 2006]
  - Precomputed avatar behavior [Lee et al, SCA 2004]
  - Responsive characters from motion fragments [McCann and Pollard, SIGGRAPH 2007]

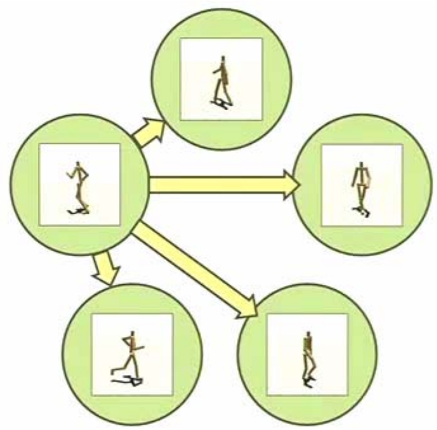
Motion

- Avoid nonsmooth blends
- Maximum blending during ground contact
- Minimum blending during flight phase

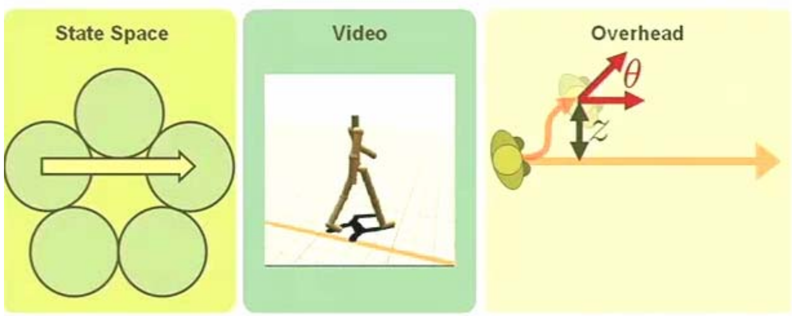




Control



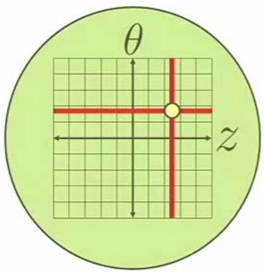
- Plan for the future
  - Real-time
- State
  - X = (C, z, θ)

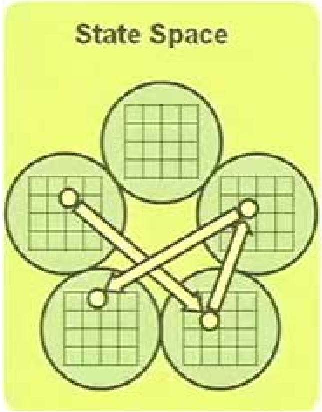


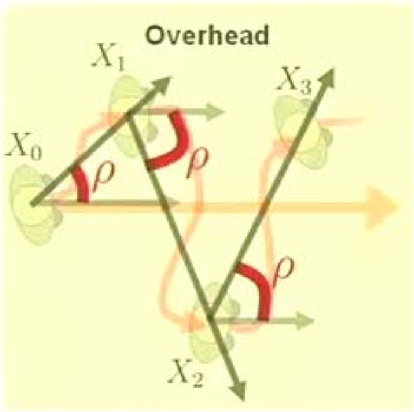
- Cost

$$c(X) = |z| \qquad c(X_i, X_j) = |\rho|$$

Value Function







$$V(X_0) = \sum_{i=0}^{\infty} \alpha^i \Big(c(X_i) + c(X_i, X_{i+1}) \Big)$$

$$V(X_0) = c(X_0) + c(X_0, X_1) + \alpha V(X_1)$$

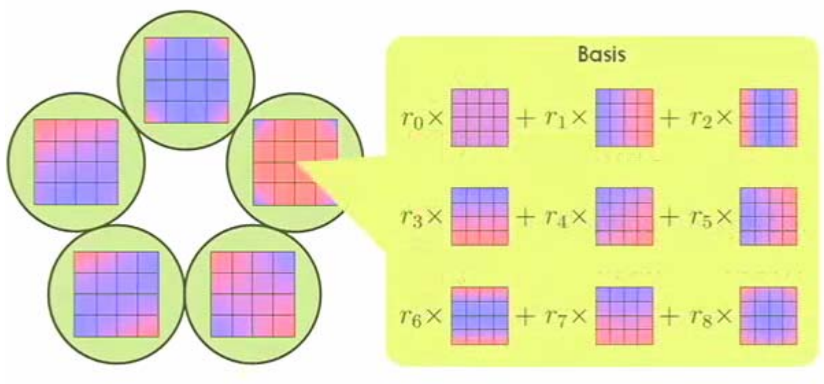
Value function

$$\Pi_\star(X) = \underset{C \in \mathcal{C}}{\operatorname{argmin}} \left\{ c_t(X, X') + \alpha V(X') \right\}$$

- Difficulties
  - Continuous space
  - Sampling is costly
  - High-dimensional controllers
    - Switching between value functions
    - Blending value functions

Near optimal policy

- A linear programming approach
  - Draw samples $\bar{\mathcal{S}} \subset \mathcal{S}$
  - Compile a set of optimal actions according to the current V from every sample in $\bar{\mathcal{S}}$ and add them as constraints $\mathcal{L} \leftarrow \mathcal{L} \cup \{(X, X') \mid X \in \bar{\mathcal{S}}\}$
  - Find V by solving the linear problem



$$\max_{\mathbf{r}} \sum_{X \in \bar{\mathcal{S}}} V(X)$$

$$\text{s.t.} \quad V(X) \leq c_s(X) + c_t(X, X') + \alpha V(X') \;\; \forall (X, X') \in \mathcal{L}.$$

Runtime control

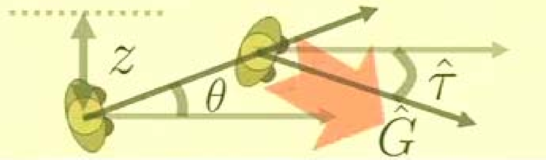
- Every time a clip finishes, scan through all clips, picking the minimum value transition

$$\Pi_\star(X) = \underset{C \in \mathcal{C}}{\operatorname{argmin}} \left\{ c_t(X, X') + \alpha V(X') \right\}$$

Examples

- Navigation

Overhead:



State:

$$X = (C, z, \theta, \hat{G}, \hat{\tau})$$

Costs:

$$c(X) = \text{torso} + \text{deviation} + \text{gait}$$

$$c(X_i, X_j) = \text{direction} + \text{blend}$$

Examples

- Spinning

Overhead:

State:

$$X = (C, \theta, \hat{\omega})$$

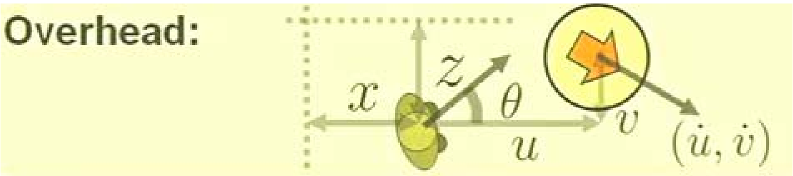
Costs:

$$c(X) = 0$$

$$c(X_i, X_j) = \text{direction} + \text{blend} + \text{spin}$$

Examples

- Obstacles



State:

$$X = (C, x, z, \theta, u, v, \dot{u}, \dot{v})$$

Costs:

$$c(X) = \text{deviation}$$

$$c(X_i, X_j) = \text{direction} + \text{blend} + \text{obstacle}$$

Conclusion

- Ease of authoring
- Dimensionality in control
- Compact basis for value function

Papers

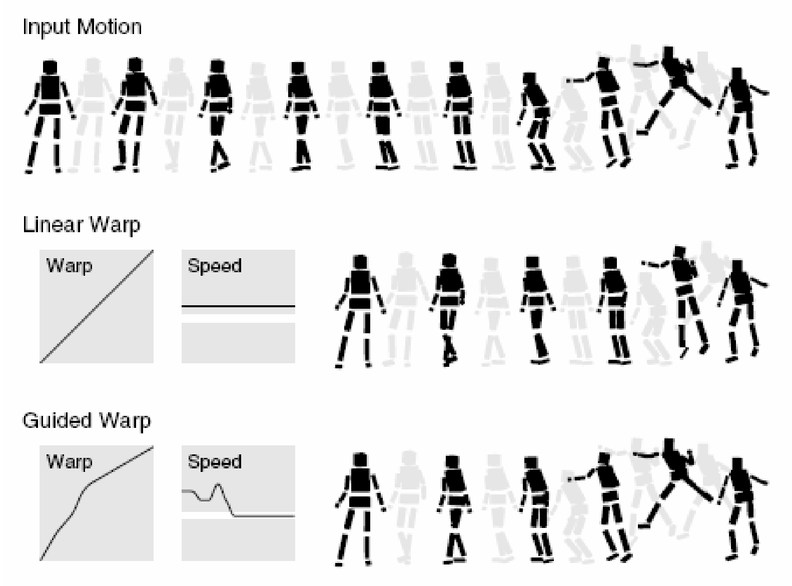
- Interpolated Motion Graphs
- Constraint-Based Motion Optimization
- Graph Optimal Control
- Guided Time Warping

Guided Time Warping

- Time warping: a tool that adjust the timing of animated character without changing poses
  - Significant manual intervention
  - Iteration of tuning and reviewing
- A new motion editing tool that simplifies the task
  - Use reference video

Overview

- A discrete optimization problem
  - Preserve the original motion and mimic the timing of the reference video
  - User-provided Constraints

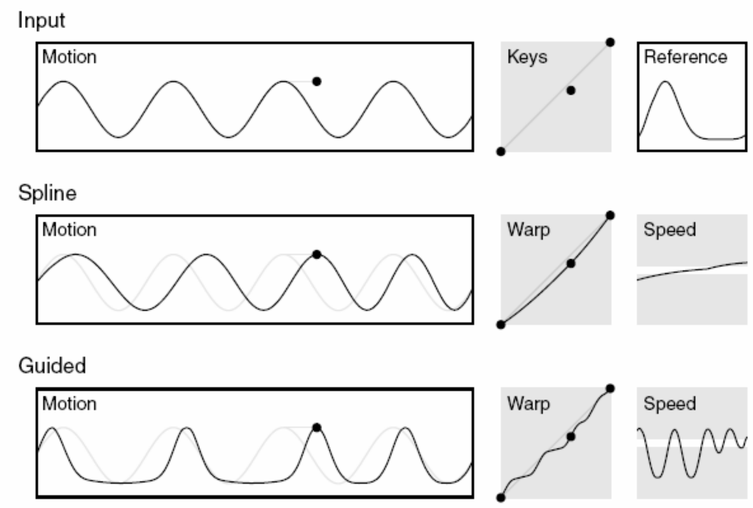


Related work

- Spline-based time warping [Witkin and Popovic 95]
- Physically accurate time warps [McCann et al 06]
- Motion blending and morphing
- Other domains
  - Rapid review of speech signals

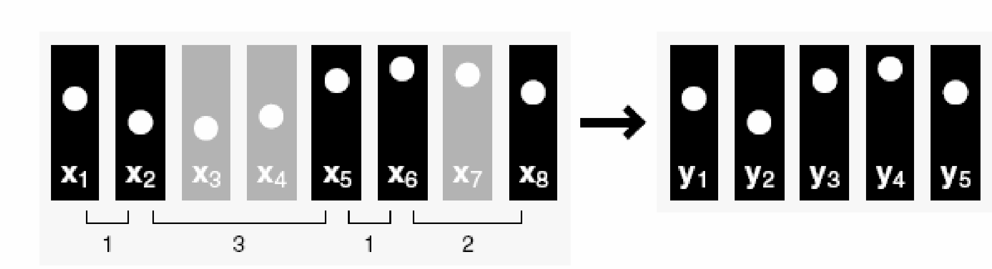
Method

- User input:
  - Motion, keyframes, a reference motion
- Dynamic time warping
- Maintain local similarities
  - Minimize object function
- Post-processing



Constraints

- Complete: entire input motion
- Monotonic: no reverse nor loop
- Time compression, modification and expansion
- keyframes



Objective

- A global correspondence between the input motion and the reference?
- Local similarity of velocities and accelerations

$$\sum_{i=2}^{m-1} f(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{y}_{i+1}). \qquad f(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{y}_{i+1}) = \frac{1}{k} \sum_{\hat{\mathbf{z}}_j \in N_i} \left\| \hat{\mathbf{y}}_i - \hat{\mathbf{z}}_j \right\|.$$

Optimization

- Graph representation
  - Vertex: a pair of input frames (x_i, x_j)
  - Edge: weighted factor $f(x_i, x_j, x_k)$
  - A time warp: a path from (x_1, x_i) to (x_j, x_n)
- Dynamic programming

$$
\begin{aligned}
c_1(\mathbf{x}_i, \mathbf{x}_j) &= 0, \\
c_p(\mathbf{x}_j, \mathbf{x}_k) &= \min_i c_{p-1}(\mathbf{x}_i, \mathbf{x}_j) + f(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k).
\end{aligned}
$$

  - $c_p(x_i,x_j)$: cost of the shortest p-edge path to vertex (x_i,x_j)

Post processing

- Discrete approximation to the optimal solution
  - Jumpy results
- Uniform smoothing
  - Important details are dampened
- Local average
  - Window size equal to the amount of compression
  - No additional modification to expansion

Conclusion

- A simple alternative to tedious manual specification
- A flexible representation of timing
- Future works
  - Little control
    - Possible solution: blend between guided warps and spline warps
  - Non-physical animation, effects