Terrain Runner: Control, Parameterization, Composition, and Planning for Highly Dynamic Motions

Libin Liu*  KangKang Yin†  Michiel van de Panne‡  Baining Guo§

*Tsinghua University  †National University of Singapore
‡University of British Columbia  §Microsoft Research Asia
Motivation

[clips are from YouTube, uploaded by 3runTube, l1consolable, ParkourGenerations, rubenparkour, traceurelements]
Outline

• Motivation

• Related work

• Controller synthesis pipeline + results

• Conclusion
Related Work

• Kinematic Methods

• Physics-based Methods
  – Single controllers: [Hodgins et al. 1995; Zhao and van de Panne 2005; Muico et al. 2009; Coros et al. 2010; Lee et al. 2010; Wang et al. 2009]

System Overview

1. single controller
   - run
   - jump
   - vault
   - drop roll

2. parameterization
   - parameterized controllers
     - run
     - jump
     - vault
     - drop roll

3. composition control

4. online planning
Motion Examples

single example motion clips

run
jump
vault
drop roll
System Overview

1. single example motion clips
   - run
   - jump
   - vault
   - drop roll

2. single controller

3. parameterized controllers
   - run
   - jump
   - vault
   - drop roll

4. composition control

5. online planning

6. parameterization
Stage 1: Single Controller Construction

1. Single controller

- Single example motion clips
- Run
- Jump
- Vault
- Drop roll

(a) Open loop policy [Liu et al. 2010]
(b) Reduced-order closed-loop policy [Ding et al. 2012]
Stage 1a: Open-loop Policy

[Liu et al. 2010]: Sampling-based Contact-rich Motion Control, SIGGRAPH 2010
Stage 1b: Reduced-order Closed-loop Policy


\[
\delta a = M \delta s + \hat{a}
\]

\[
\delta a = a - \tilde{a}
\]

\[
s - \tilde{s} = \delta s
\]
Stage 1b: Reduced-order Closed-loop Policy

\[ \delta a = M \delta s + \hat{a} \]

- Change in control
- Change in states

\[ M_{ap} \cdot M_{sp} \]

\[ s - \bar{s} = \delta s \]

\[ \delta a = a - \bar{a} \]
Stage 1b: Reduced-order Closed-loop Policy

\[ \delta a = M \delta s + \hat{a} \]

\[ M_{ap} \cdot M_{sp} \]

change in control

change in states

reduced-order state
Stage 1b: Feedback Policy

Manually-selected States: s

- Running: 12 dimensions
Stage 1b: Feedback Policy

Manually-selected Controls: a

- for all skills: 9 dimensions
Stage 1b: Feedback Policy

Multi-phase Skills

• Vaulting
Stage 1b: Feedback Policy

Multi-phase Skills

- Drop-rolling

phase 1: jumping  phase 2: dropping  phase 3: rolling  phase 4: standing-up
Stage 1b: Feedback Policy

Manually-selected States: $s$

- Jumping

- Vaulting

- Drop-rolling
Stage 1b: Feedback Policy

Optimization

\[ \delta a = M \delta s + \hat{a} \]

- Optimize \( M \)
  - CMA, Covariance Matrix Adaption ([Hansen 2006])
  - Running:
    - Objective function
      \[ E = w_t (N_d T_c - T_S) + w_s E_s + w_p E_p + w_\tau E_\tau \]
    - 12 minutes on 24 cores
    - more details in paper and [Ding et al. 2012]
System Overview

1. single example motion clips
   - run
   - jump
   - vault
   - drop roll

2. single controller

3. parameterized controllers
   - run
   - jump
   - vault
   - drop roll

4. composition control

Parameterization

Online planning
Stage 2: Parameterization

\[ \delta a = \begin{bmatrix} M_\theta \end{bmatrix} \delta s + \hat{a}_\theta \]

\[ \delta a = \begin{bmatrix} M \end{bmatrix} \delta s + \hat{a} \]
Stage 2: Parameterization

Running: parameter space

- $\theta = (v, \phi)$
  - speed, turning rate
  - $[2\text{m/s}, 5\text{m/s}] \times [-6^\circ, 6^\circ]/\text{step}$
Stage 2: Parameterization

Running: Action Set Augmentation

- $\hat{a}_\theta = \{ l, r, \alpha, \beta \}$

- Space scaling and time scaling diagrams.
Stage 2: Parameterization

Running: optimization

- $M_\theta, \hat{a}_\theta$

\[
E = w_t (N_d T_c - T_s) + \frac{w_h}{T_s} \int \|d_h - \bar{d}_h\| dt
\]

\[
\text{success} + \frac{1}{N_s} (w_\phi |\phi_i - \phi^*| + w_v |v_i - v^*| + w_f |f_i - f^*|)
\]

\[
\text{head's stability}
\]

desired parameters
Stage 2: Parameterization

Continuation

• [Yin et al. 2008]: Continuation methods for adapting simulated skills. SIGGRAPH 2008

\[ f(\theta) = p(\theta) + \sum_{i=1}^{N} \lambda_i \varphi(|\theta - \theta_i|) \]
Stage 2: Parameterization

Continuation

• Predictor-corrector
Stage 2: Parameterization

Running Results

Run - (0°, 2.0 m/s)
Stage 2: Parameterization

Obstacle Clearing Maneuvers

- $\theta = h$
  - Obstacle height
    - Jumping – [0.1m, 0.7m]
    - Vaulting – [0.6m, 1.0m]
    - Drop-rolling – [0.9m, 2.0m]
Stage 2: Parameterization

Obstacle Clearing Maneuvers

- Optimization

\[ \hat{a}_h = \{ \text{contact, balance, pose} \} \]

\[ E_h = w_c E_c + w_b E_b + w_p E_p \]
Stage 2: Parameterization

Obstacle Clearing Maneuvers

- Jumping
  - Contact term $E_c$
Stage 2: Parameterization

Obstacle Clearing Maneuvers

- Vaulting
  - Contact term $E_c$
Stage 2: Parameterization

Obstacle Clearing Maneuvers

- Drop-rolling
  - Balance term $E_b$
Stage 2: Parameterization

Obstacle Clearing Results

Jump - 10cm
System Overview

1. single example motion clips
   - run
   - jump
   - vault
   - drop roll

2. single controller

3. parameterized controllers
   - run
   - jump
   - vault
   - drop roll

4. online planning

composition control
Stage 3: Composition

Three-phase Composition Scheme

\[ \delta a = M'_{\theta_{in}} \delta s + \hat{a}_{\theta_{in}} \]

\[ \delta a = M_{\theta_{out}} \delta s + \hat{a}_{\theta_{out}} \]

Obstacle
Stage 3: Composition

Three-phase Composition Scheme

Three-phase Composition Scheme

run
Stage 3: Composition Optimization

- Parameters
  \[
  \{\theta_{in}, M'_{\theta_{in}}, M'_h, S_{\theta_{out}}\}
  \]

Three-phase Composition Scheme

\[\delta a = M'_h \delta s + \hat{a}_h\]

\[\delta a = M'_{\theta_{in}} \delta s + \hat{a}_{\theta_{in}}\]

\[\delta a = S_{\theta_{out}} \delta s\]
Stage 3: Composition

Results

Jump - 30cm
System Overview

1. single example motion clips
   - run
   - jump
   - vault
   - drop roll

2. single controller
   - parameterization

3. parameterized controllers
   - run
   - jump
   - vault
   - drop roll

4. composition control
   - online planning
Stage 4: Online Planning

- Step-based kinematic planning

\[
\begin{align*}
\min w_u (u - u_n)^2 + \sum w_i (v_i - v_{i-1})^2 \\
\text{s.t. } v_n = v_{in}
\end{align*}
\]
Stage 4: Planning

Results
Results

Terrain Running
Conclusion

• Parkour-style motions
  – Running, jumping, vaulting, drop-rolling

• Complete framework, Realtime synthesis
  – Control construction, parameterization, composition, planning

• Structured optimization scheme
Limitations

- Only partly automated
- Composition can fail
- No arbitrary transitions
Future Work

[Parkour Memories, uploaded by 3runTube
http://www.youtube.com/watch?v=24cgnAA6x0l&hd=1]
Thanks