Phase-Functioned Neural Networks for Character Control

Dennis Sprokholt & Mark de Jong
Motivation

- Character movement animations
  - Responsive
  - Rough terrain
Problem

- General:
  - Create responsive motions
- Specific:
  - Data-Driven
  - This means:
    - Large training set (motions & terrains)
    - Realtime
    - Low memory requirement
    - Little manual preprocessing
Contribution

- Phase-Functioned Neural Network (PFNN)
  - NN with phase parameter

- Terrain fitting of locomotion data
Related Work

- Data-Driven Locomotion Synthesis
  - PCA - Limited motion variety
  - Kernel-based - High memory cost
- Environment Interaction
  - Not scalable
- Mapping User Parameters to Latent Variables
  - Variational autoencoder
Terrain Fitting

Technical Details

- Raytrace Heightmaps
  - 1 px/inch

- Sample 20,000 patches sized 3x3m

- 10 best patches per locomotion cycle
  - Minimizing fitting error
    \[ E_{fit} = E_{down} + E_{up} + E_{over} \]
  - Make feet touch floor
Phase-Functioned Neural Networks

Technical Details: Input

- Joint Positions
- Joint Velocities
  (Previous Frame)

Input:
- Trajectory Locations
- Trajectory Directions
- Trajectory Heights

Output:
- Joint Positions
- Joint Velocities
- Joint Rotations
- etc...
Phase-Functioned Neural Networks

Technical Details: Input

Trajectory:
- 12 samples
- 1s in past
- 0.9s in future

Locations

\[ t_p^i \in \mathbb{R}^{2t} \]
Phase-Functioned Neural Networks

Technical Details: Input

Trajectory:
- 12 samples
- 1s in past
- 0.9s in future

Directions

\[ t^d_i \in \mathbb{R}^{2t} \]
Phase-Functioned Neural Networks

Technical Details: Input

Trajectory:
- 12 samples
- 1s in past
- 0.9s in future

Heights

$$t^h_i \in \mathbb{R}^{3t}$$
Phase-Functioned Neural Networks

Technical Details: Output

- Joint attributes
  - Positions
  - Velocities
- Root transform translational velocity (x,z)
- Character rotational velocity
- Change in phase
- Foot contact labels
Phase-Functioned Neural Networks

Technical Details

- 4 layered NN
  - Input
  - 2 hidden layers
  - Output
- Phase function: cubic Catmull-Rom Spline
- 4 control points consisting of NN weight configurations
Training

Technical Details

- Scale joint values down
- Per locomotion clip, train on selected patches
- Basically an optimization problem
- 30 hours on gtx 660

\[ \text{Cost}(X, Y, P; \beta) = \|Y - \Phi(X; \Theta(P; \beta))\|^2 + \gamma|\beta| \]
Results
Results
Evaluation

- Standard neural net
- Encoder-Recurrent-Decoder network
- Gaussian Process approach

<table>
<thead>
<tr>
<th>Technique</th>
<th>Training</th>
<th>Runtime</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFNN cubic</td>
<td>30 hours</td>
<td>0.0018s</td>
<td>10 MB</td>
</tr>
<tr>
<td>PFNN linear</td>
<td>30 hours</td>
<td>0.0014s</td>
<td>25 MB</td>
</tr>
<tr>
<td>PFNN constant</td>
<td>30 hours</td>
<td>0.0008s</td>
<td>125 MB</td>
</tr>
<tr>
<td>NN (a) (b)</td>
<td>3 hours</td>
<td>0.0008s</td>
<td>10 MB</td>
</tr>
<tr>
<td>ERD (c) (d)</td>
<td>9 hours</td>
<td>0.0009s</td>
<td>10 MB</td>
</tr>
<tr>
<td>GP (e)</td>
<td>10 minutes</td>
<td>0.0219s</td>
<td>100 MB</td>
</tr>
<tr>
<td>PFGP (f)</td>
<td>1 hour</td>
<td>0.0427s</td>
<td>1000 MB</td>
</tr>
</tbody>
</table>
Critical Analysis - The Good

- Well structured
- Realistic resulting locomotion
- Wide range of applications and resulting motions
- Locomotion suitable to the terrain
Critical Analysis - The Bad

- Long and complicated sentences
- Not all choices are explained well
- Training and terrain fitting details missing
  - Amount of MoCap/training data
  - FPS of MoCap data
- Long training time
Future Work

- Obstacles in terrain
- Combine with IK
- Speed up training
- Predictable results
Thanks

Questions?