Autonomous Robot Dancing Driven by Beats and Emotions of Music

Gus (Guangyu) Xia
Junyun Tay
Roger Dannenberg
Manuela Veloso

Carnegie Mellon University,
AAMAS2012
Introduction: Goal

- Context: Many robot dances are preprogrammed.
- Idea: How about automating the task?
- Our goal:

  Given a piece of music, we want to automatically generate a robot dance.

  The dance should be interesting, safe, reflecting the emotion, and synchronized to the beats of music.
Introduction: General approach

Input:
Music Audio

Music Content Analysis

Beats & Emotion

Motion Primitives

Parameterized & labeled Keyframe sequences

Dancing Plan

Output:
Robot Dance

Notice: the dancing plan is created offline, and then executed and synchronized with music audio.
Outline

- Introduction
- Motion primitives
- Music content analysis
- Dancing plan
- Execution
- Demo
- Conclusion
Motion primitives (MPs)

- Robot we use
  - NAO
  - Stand-alone autonomous robot
  - 21 joints
- Four catalogs: Head, Left Arm, Right Arm, Legs
- For each catalog, we build its own MPs library
Motion primitives: properties

- Large variety of combinations
- Allow speed change
- Safe to execute
- Convey emotions
Motion primitives: Variety

- In order to maximize variety of the combination of Motion Primitives:
  - Design the library for four catalogs *independently*
    - Head: 8, Left Arm: 9, Right Arm: 9, Legs: 26
  - Execute the four catalogs *simultaneously*
  - Though we only designed 52 MPs in total, there are thousands of possible combinations at each time while execution.
Motion primitives: Parameterization

- Motion primitive = sequence of keyframes

\[
\begin{bmatrix}
K_1 & K_2 & K_3 & K_4 & K_5 \\
\hline
t_1 & t_2 & t_3 & t_4 & t_5
\end{bmatrix}
\]

- In order to allow speed change:
- A single stretching parameter $\beta$
- If $t_1 = 0$, the parameterized motion primitive is

\[
\begin{bmatrix}
K_1 & K_2 & K_3 & K_4 & K_5 \\
\hline
t_1 & \beta t_2 & \beta t_3 & \beta t_4 & \beta t_5
\end{bmatrix}
\]
Motion primitives: Safety

- In order to make sure MPs are safe execute:
  - Define Minimum Interpolation Time (MIT) between keyframes and force execution time larger than MIT

- Case 1: Within a MP:
  - MIT is designed as difference between two contiguous time stamps, so that $\beta$ is no less than 1

- Case 2: Between two MPs:
  - MIT is from the last keyframe of current MP to the first keyframe of next MP: $dist_M(M_n, M_{n+1})$
Motion primitives: Emotion Label

- In order to get conveyed emotions:
- Automatically label the emotion of MPs based on a pre-labeled keyframe library
- For each keyframe
  - ...find nearest static pose in our pre-labeled library, return their emotions
- Use a weighted sum of emotions to estimate emotion of the motion primitive
Example: Static Postures Vs. MPs

<table>
<thead>
<tr>
<th>Static Posture</th>
<th>Selected Keyframe in Motion Primitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Happy</td>
<td></td>
</tr>
<tr>
<td>Sad</td>
<td></td>
</tr>
<tr>
<td>Angry</td>
<td></td>
</tr>
</tbody>
</table>

Selected Keyframe in Motion Primitive
Music Content Analysis

■ We want to extract emotion and beat times from music, to use as cues for the robot dance.
■ MPs should convey the emotions of music
■ MPs should be synchronized with the beat of music
Music Content Analysis: Emotion

Input: Music Audio

Feature Extraction

Support Vector Regression (SVR)

2D Representation (based on Thayer)

Valence

(positive)

Peaceful  Relax  Pleased  Happy

Sleepy  Calm  Excited

(negative)

Sad  Bored  Nervous  Angry

(activation)

(low)

(high)

Byeong-jun Han, et al. (2009) SMERS: Music Emotion Recognition by using SVR

One more step: use a 30-second sliding window to compute the trajectory of music emotion
A/V value of emotions on the 2-D plane

<table>
<thead>
<tr>
<th>Emotion</th>
<th>Activation</th>
<th>Valence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Happy</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Sad</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>Angry</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>Surprised</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Fear</td>
<td>0.5</td>
<td>-1</td>
</tr>
<tr>
<td>Disgust</td>
<td>-0.5</td>
<td>-0.5</td>
</tr>
</tbody>
</table>

Metric of emotion similarity: $dist_E(E(M), e)$
- The Euclidian distance on the 2-D plane
- $E(M)$ : emotion of the motion primitive $M$
- $e$ : emotion of the music
Music Content Analysis: Beat

Input:
Music Audio

Feature Extraction

Periodicity estimation using autocorrelation
Search for roughly equally spaced peaks using DP

Dan Ellis (2007), Beat Tracking by Dynamic Programming
Dancing Plan

- We have got music information and MPs
- Combine them to generate a sequence of MPs
- Nondeterministic, Smooth, Emotional, Synchronized

Solution: Sampling from a stochastic process

A generative model, sequentially generating MPs by drawing samples from $p(M_{n+1} | M_n, e_{n+1})$
Dancing Plan: To define \( p(M_{n+1} \mid M_n, e_{n+1}) \)

- Nondeterministic
- Smooth:
  - Continuity from one motion primitive to the next
  - Continuity Factor: 
    \[
    CF = \frac{1}{\sqrt{2\pi \sigma^2_M}} \exp\left\{ -\frac{\text{dist}^2_M(M_n, M_{n+1})}{2\sigma^2_M} \right\}
    \]
- Emotional:
  - Considering the music, MPs should reflect the emotion of music.
  - Emotion Factor: 
    \[
    EF = \frac{1}{\sqrt{2\pi \sigma^2_E}} \exp\left\{ -\frac{\text{dist}^2_E(E(M_n), e_n)}{2\sigma^2_E} \right\}
    \]
- Definition: 
  \[
  p(M_{n+1} \mid M_n, e_{n+1}) = CF \cdot EF \cdot N
  \]
- Synchronized: stretch selected MP, making its last keyframe end on a beat time
Dancing Plan: Review

For each catalog, we iteratively:

- Get the detected emotion at the end of current motion primitive, $e_{n+1}$
- Draw a new motion primitive, $M_{n+1}$, from the generative distribution $p(M_{n+1} | M_n, e_{n+1})$
- Stretch $M_{n+1}$ to end on nearest future beat time
Execution

- Even if the planned timing is perfect, there are latency and other execution time errors
- Solution: Real time synchronization algorithm to overcome time drifting
- At each step while execution, iteratively:
  - Check the timing and then re-schedule next step
Video Demo
Conclusion and acknowledgements

Conclusion

- An approach to automate robot dancing, based on matching parameterized MPs to music features
- Nondeterministic, Smooth, Emotional, and Synchronized with music
- A complete demonstration with a NAO humanoid robot with multiple pieces of music.
- The scheme generalizes to other robots

Acknowledgements

- Byeong-jun Han
- Somchaya Liemhetcharat
Synchronized with music beats

- Just stretch the selected motion primitive, making its last keyframe ending on a beat time.
- Make sure:
  - Interval is no less than $\text{dist}_M(M_n, M_{n+1})$
  - $\beta$ is no less than 1