

Plan2Dance: Planning Based Choreographing from Music

Yuechang Liu,¹ Dongbo Xie,¹ Hankz Hankui Zhuo,^{2*} Liqian Lai¹

¹Jiaying University, China. {ycliu, 201701041}@jyu.edu.cn, sfreebobo@163.com

²Sun Yat-sen University, China. zhuohank@mail.sysu.edu.cn

Abstract

The field of dancing robots has drawn much attention from numerous sources. Despite the success of previous systems on choreography for robots to dance with external stimuli, they are often either limited to a pre-defined set of movements or lack of considering “hard” relations among dancing motions. In the demonstration, we design a planning based choreographing system, which views choreography with music as planning problems and solve the problems with off-the-shelf planners. Our demonstration exhibits the effectiveness of our system via evaluating our system with various music.

Introduction

The field of dancing robots has drawn much attention from numerous sources. For example, SONY presented a humanoid robot, called QRIO (Geppert 2004), which can dance with multiple units in a highly coordinated fashion via imitating human’s dance. Nakaoka et al. explored a motion capture system to teach a robot, called HRP-2, a traditional Japanese folk dance (Nakaoka et al. 2005). Despite the success of previous systems, they are often either limited to a pre-defined set of movements (together with music) or demonstrate little variance according to external stimuli. To improve the variance of dances, Bi et al. proposed to allow legged robots to dance in synchronization with musics in a diverse fashion (Bi et al. 2018). They created a dance choreography by picking from a dance motion library according to the beats of music. Dance motions include various stepping and base motions. The picking process from the library is defined by a Markov chain, which depends on the previously picked dance motion and the current music tempo. In those probabilistic graphical models based approaches, motions which are unreasonable to previous motions, are often selected, due to the limit of probabilistic models on representing logical relations between dancing motions.

In this demonstration, we design a system, called Plan2Dance, to create choreography based on music. A set of action models is built (in the form of PDDL (Fox and Long 2003) language) based on the relations of basic dancing motions, by considering the temporal requirement of

dancing motions and choreography preferences according to music. We convert each input music into a planning problem and solve the problem with an off-the-shelf planner based on the created action models.

The challenges of the demonstration are two folds. The first one is about how to build PDDL action models based on basic dancing motions, which involves taking care of complex domain knowledge of dancing (we created 58 action models based on a library of basic dancing motions). The other one is about how to convert a music into a planning problem (including an initial state and the goal).

An Overview of Plan2Dance

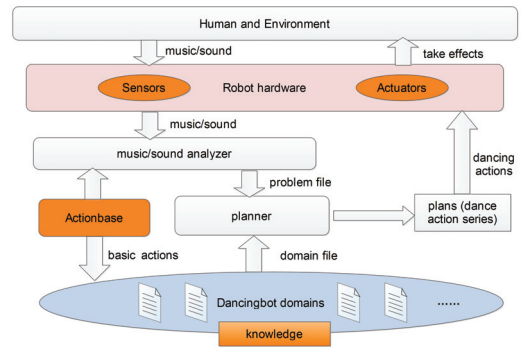


Figure 1: An overview of our Plan2Dance system

We exhibit an overview of our Plan2Dance system in Figure 1. We first take music, i.e., sound waves, as input from environments via robot’s sensors. We then analyse the music based on basic actions of dances (denoted by “Actionbase” in Figure 1) using machine learning approaches (we exploited the pyAudioAnalysis library (Giannakopoulos 2015) for building planning problems from music), and output a planning problem, including an initial state and a goal, for an off-the-shelf planner. Note that we collect a library of basic actions of dances, i.e., “Actionbase”, and create their action models, namely “Dancingbot domains” in Figure 1, according to the relations of dancing motions with the help of dancing experts. Provided as input the planning problem, denoted by “problem file” in Figure 1, and action models, denoted by “domain file” in Figure 1, an off-the-shelf planner

*corresponding author

(we exploit OPTIC-CLP (Benton, Coles, and Coles 2012) in our system) calculate a solution plan (denoted by “plans (dance action series)” in the figure) and feed the plan to the robot. The robot sequentially executes the dancing plan with its actuators.

Building action models and planning problems Figures 2 and 3 show examples of planning problems and action models built from music and dancing experts. Figure 2 is an example planning problem including three objects indicating moving speed of the robot, “slow”, “mid” and “fast”, an initial state and a goal, where preferences and plan metric are used for specifying dance duration according to the specific music. Figure 3 demonstrates two temporal action “handsWaveDown” and “dumbAction”, where numeric and temporal features are used for defining relations among actions and temporal constraints of dances. Detailed description of all action models and planning problems with respect to specific music can be found from the online website¹.

```
(define (problem robot-1)
  (:domain DancingRobot)
  (:objects slow - rate mid - rate fast - rate )
  (:init (is_body_free) (= (dance-time) 0) (= (dumb-time) 0)
    (= (action-rate slow) 1.15) (= (action-rate mid) 1)
    (= (action-rate fast) 0.85) (= (dance-total-time) 11.75)
    (= (dumb-prop) 0.1) (= (dumb-time) 1.5) )
  ;;-----other init states-----
  (:goal (and (> (dance-time) 11.0) (< (dance-time) 12.5)
    (forall (?s - state) (preference p0 (best-rate ?s fast)))
    (preference p1 (>= (/ (dumb-time) (dance-total-time)) (dumb-prop))))
  (:metric minimize (* (is-violated p0) (is-violated p1))))
)
```

Figure 2: An example problem built from music.

<pre>(:durative-action handsWaveDown :parameters (?rate - rate) :duration (= ?duration (* 2.0 (action-rate ?rate))) :condition (and (at start (and (at_space_y y_mid) (at_space_z z_up) (at_hand others) (is_body_free)))) :effect (and (at start (and (not (is_body_free)) (not (at_space_y y_mid)) (not (at_space_z z_up)) (not (at_hand prethoracic)))) (at end (and (increase (dance-time) (* 2.0 (action-rate ?rate))) (is_body_free) (finished s6) (best-rate s6 ?rate) (increase (action-times s6) 1) (at_space_y y_mid) (at_space_z z_up)))))</pre>	<pre>(:durative-action dumbAction :parameters () :duration (= ?duration (dumb-time)) :condition (and (at start (and (is_body_free)))) :effect (and (at start (and (not (is_body_free)))) (at end (and (increase (dance-time) (dumb-time)) (increase (dumb-total-time) (dumb-time)) (is_body_free)))))</pre>
--	--

Figure 3: Two example action models built based on relations of dancing actions.

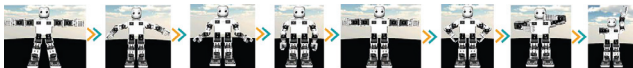


Figure 4: A demonstration of a dancing plan

Experiment

Plan2Dance is implemented in Python, running on Ubuntu Linux 18.0 with virtual machine of AMD CPU 2.0G and DDR3 6G RAM. The humanoid robot - ROBOTIS MINI from ROBOTIS is adopted to demonstrate dancing actions. We adopted a music library with 10 modern musics and constructed a motion library with 58 basic motions². An example dancing result is shown in Figure 4.

Conclusion

We propose Plan2Dance, a system to automatically choreograph dances given a music file input. We build planning problems and action models based on the analysis of the input music and dancing experts and calculate plans for robots to dance automatically using an off-the-shelf planner. In the future it would be interesting to study learning action models (Zhuo and Yang 2014; Zhuo, Muñoz-Avila, and Yang 2014) and planning with the learnt models (Zhuo and Kambhampati 2017) automatically from dances demonstrated by dancing experts.

Acknowledgement

We thank the support of NFSC (U1611262), Guangdong Natural Science Funds for Distinguished Young Scholar (2017A030306028), Guangdong special branch plans young talent with scientific and technological innovation, Pearl River Science and Technology New Star of Guangzhou.

References

Benton, J.; Coles, A.; and Coles, A. 2012. Temporal planning with preferences and time-dependent continuous costs. *ICAPS*.

Bi, T.; Fankhauser, P.; Bellicoso, D.; and Hutter, M. 2018. Real-time dance generation to music for a legged robot. In *IROS*, 1038–1044.

Fox, M., and Long, D. 2003. PDDL2.1: an extension to PDDL for expressing temporal planning domains. *J. Artif. Intell. Res.* 20:61–124.

Geppert, L. 2004. Qrio, the robot that could. *IEEE Spectrum* 41(5):34–37.

Giannakopoulos, T. 2015. pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. *PLOS ONE* 10(12):1–17.

Nakaoka, S.; Nakazawa, A.; Kanehiro, F.; Kaneko, K.; Morisawa, M.; and Ikeuchi, K. 2005. Task model of lower body motion for a biped humanoid robot to imitate human dances. In *IROS*, 3157–3162.

Zhuo, H. H., and Kambhampati, S. 2017. Model-lite planning: Case-based vs. model-based approaches. *Artif. Intell.* 246:1–21.

Zhuo, H. H., and Yang, Q. 2014. Action-model acquisition for planning via transfer learning. *Artif. Intell.* 212:80–103.

Zhuo, H. H.; Muñoz-Avila, H.; and Yang, Q. 2014. Learning hierarchical task network domains from partially observed plan traces. *Artif. Intell.* 212:134–157.

¹<http://xplan-lab.org/plan2dance/>.

²A video demonstration of Plan2Dance’s performance can be found at <https://www.bilibili.com/video/av68454866/>