Integrating an autonomous robot on a dance and new technologies festival

P. Abad[1], M. Franco[1], R. Castillón[1], I. Alonso[1], A. Cambra[1], J. Sierra[2], L. Riazuelo[1], L. Montano[1], A. C. Murillo[1]

[1] Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain.
[2] Centro Universitario de la Defensa. Zaragoza, Spain

Abstract. This paper presents the results of a project to integrate an autonomous mobile robot into a modern dance performance at a dance and new technologies festival. The main goal is to integrate a simple low cost mobile robot into the dance performance, in order to study the possibilities that this kind of platforms can offer to the artists. First, this work explains the process and design to embed the robotic platform into the choreography theme. Another contribution described in this work is the system architecture proposed and built to make the robot behaviours match the artists requirements: precise, synchronized and robust robot movements. Finally, we discuss the main issues and lessons learned for this kind of robotics and arts applications and summarize the results obtained, including the successful final live performance results.

Keywords: Autonomous robots. Arts. Modern Dance.

1 Introduction

Over the last years we have witnessed the introduction of robotics into many diverse fields. There are numerous recent attempts to integrate new AI & robotics related technologies on novel areas of application, and the Art fields are not an exception. There are many recent initiatives to study how Arts could integrate and benefit from robotics: from more generic arts and education applications [4], to approaches to create paintings made by robots [9], to build interactive sculptural systems [2] or to experiment with robotic orchestras [7]. Our work explores the possibilities to integrate a robotic member into a modern dance team.

The main goal of this experimental project is to study the possibilities that a low cost mobile robot can provide artists involved on a performance. This paper presents the results of integrating an autonomous mobile robot in a modern dance festival performance. The festival, Trayectos[3], was held in a public Art & Technology Center (Etopia) in Zaragoza, and the dance company involved in this project is *Tarde o Temprano Danza*[4].

One of the challenges that makes this project original is the goal of making the robot one member of the dance team. The dancers require the robot to have

[3] http://www.danzatrayectos.com/en/laboratorio-de-danza-y-nuevos-medios/
[4] http://tardeotempranodanza.wix.com/tardeotempranodanza

Fig. 1. This project presents a multidisciplinary experiment at an *Arts and Technology* center to integrate a TurtleBot robot into a dance and new technologies festival.

a pre-defined and accurate behaviour, since the performance is not an improvisation from their side. This means that the robot *dancer* should also *learn* its part of the choreography, to be perfectly synchronized with the rest of the team. One of the most important lessons learned during this project is that common theoretical and technical problems in robotic research are often not the main challenges faced in this type of application. The requirements and expectations from the non-robotics teams of this multi-disciplinary projects implied strong restrictions in aspects such as: very high accuracy of certain movements, restrictions on the kind of sensors that can be used, reactive behaviours that were not acceptable or limitations on the artificial markers or elements that can be added to the scenario or dancers outfit.

The results and contributions presented in this work are:

- The proposed **architecture**, software and hardware modules, to achieve the different robot behaviours following the dance company requirements.
- The design of a **housing** to cover the base robotic platform (TurtleBot 2) in order to embed it into the choreography theme.
- Details of the final robot choreography and **results** of the **live performance** at the festival *Trayectos* 2017.
- Discussion of **lessons learned** for this kind of robotics and Arts application.

1.1 Related Work

As previously mentioned, we find multiple attempts to integrate robotics in Art fields, not only to study what Art can gain from robotics, but also the other way around. There are recent research publications compiling different experiments on robotics and art experiments [6] and specific sessions on top robotics research venues[5]. We can find related work from top universities involved in professional artistic venues[6], or dissemination actions involving different Arts and robotics experiments and venues[7] for educational purposes [10].

Robots and Dance. Works related to performing Arts are very relevant to our project, in particular dance and theater. We can find earlier studies that were focused on the possibilities of expressing emotion and intention trough the robot body movements [11]. Among more recent works, we find researchers that propose autonomous systems that are able react to different sounds and music, adapting for example to the music tempos or motion restrictions [1][12]. There are also many studies that consider robotic dance as a tool for therapy activities and medical applications, such as the proposed work for therapy with children [14]. More details and examples of robotics and dance applications can be found on the survey work in [13]. Other approaches closer to our goals study how to integrate robots on a dance activity or performance. There are researchers studying how to design a robotic dance partner, analyzing the human-robot coordination [8][15]. They propose a system that reacts to physical interactions thanks to the use of an omni-directional mobile base equipped with force sensors. There is little experimental prior work focused on the artist requirements to design the robot choreography as part of the artists performance [3]. The focus on integrating the robot as one member of a hybrid human-robot team is a key component in our work. Differently from most prior works on dance, we have targeted our work to evaluate the reach of a low cost platform, in collaboration with local artists, making emphasis on the robot being one more member in the dance team. This goal has highlighted the need of certain capabilities in the robot such as accuracy and exact time synchronizations, rather than reactive behaviours and improvisation from more autonomous robotic platforms in prior work.

Autonomous robot navigation. Additionally, essential to this project is related work in the areas of mapping and localization. We have used well known approaches for 2D mapping [5] and particle filter based localization [16], using a 2D scanning sensor mounted on a mobile robot, as detailed in the following sections. These algorithms enable us to robustly run the dance performance, even with cluttered background, large occlusions to robot sensor field of view and very heterogeneous lighting and clutter conditions across executions.

2 Robot Design

This section describes the design and main components of the robotic platform used in this project. To help convey the message of the artists for the performance, about the mutual influence between robot and human, a housing was

[5] http://www.roboticart.org/iros2017/
[6] http://artpower.ucsd.edu/dancing-robot-huang-yi-kuka/
[7] https://www.robofest.net/index.php/current-competitions/graf

(a) (b) (c)

Fig. 2. (a) Original Turtlebot 2 available; (b) different renders of housing designed in this project; (c) 3D printed pieces models, including housing parts and support pieces to hold sensors and LEDs.

developed for the available robot model, TurtleBot 2[8]. Fig. 2 shows a rendered image of the original robot, together with three different rendered points of view of the robot equipped with the designed housing.

Robot Housing. The housing was designed and manufactured in collaboration with the choreography team to establish a common conceptual idea. This idea emerged in a creative session that took place at the beginning of the project, lead by the design specialists of the team. It served so that the different parts of the project followed the same line of work with some common final objectives towards the final demo at the dance festival. The design of the housing was strongly influenced by some restrictions inherent to the robotic platform. The most significant ones are:

– The size of the open hole around the laser sensor, to allow its operation.
– The maximum weight restriction of 5kg on the base.
– The need to leave easy access to charging and device ports.

Additionally, the artists and choreographers from the dance company had strong preferences to make the robot look *alive* and cute for the public. This limited the position of robot *face* and eyes and encouraged the possibility to simulate *breathing*. One of the main challenges at this step was to give unity to all these different requirements and restrictions. As it can be seen in Fig. 2 (b), besides the main housing (head built from 3D printed pieces and body cover consisting of a translucent and light cylinder), there were multiple small additions to give a clean and modern appearance, such as the eyes area transparent cover and multiple stickers to make the look as homogeneous as possible. Fig. 1 shows the final appearance of the robot.

Sensors and actuators. In addition to the robot housing, several additional pieces were developed to serve as support elements, both for the housing itself and for other internal elements and sensors. The vast majority of these parts have been manufactured with a 3D printer, and therefore need to be split into pieces that

[8] http://www.turtlebot.com/turtlebot2/

(a) (b) (c) (d) (e)

Fig. 3. (a-d) Different stages of the final appearance of the robot. (e) Main components, scanning laser rangefinder and LED stripes, used in the final design.

fit the measurements of the printing area of the 3D printer available. Fig. 2(c) shows the 3D model of all these support pieces, including head cover pieces, internal holds for sensors and other small components. Fig. 3 shows different stages of the robot as the different components were built and attached.

Different sensors were considered during the initial stages of this project, but many were discarded mainly for development time and weight restrictions or robustness and accuracy requirements. Besides the basic TurtleBot on-board sensors, as shown in Fig. 3d, we added a Hokuyo laser scan (since it was the most robust choice to map the environment give the requirements and restrictions of the scenario) and programmable LED stripes (since they were the only available additional component to give expressive capabilities to the system that fit the scenario and artists requirements and restrictions). Finally, we considered adding additional moving parts to the robot that had to be discarded due to weight limits on the base platform.

3 System Description

This section describes the main modules of the proposed system: 1) the modules used to map the environment where the robot is going to perform its tasks; 2) the system architecture built to program all the mobile robot tasks. All the modules are integrated and based on ROS (Robot Operating System).

3.1 Modeling and localizing the robot in the environment

Requirements and Set up. The performance scenario is an open space rather than a conventional scenario (more details in next Section). This means there is no physical separation between the public and the performance space. To ensure certain static geometric components that the robot can easily incorporate into the model (map) of the working environment we added small walls, following these requirements from the artists:

1. To follow the aesthetic requirements of the designers.
2. To avoid any unnecessary occlusions to the public during the performance.

As a low cost solution, a few expanded polystyrene blocks (40 cm tall) were made and arranged throughout the performance space. In Fig. 4(a) we can see some of these blocks (white small panels) during training sessions of the choreography.

(a) (b)

Fig. 4. (a) Images from training sessions where we can see the panels used to limit the robot map of the performance space. (b) Sample map from the performance area.

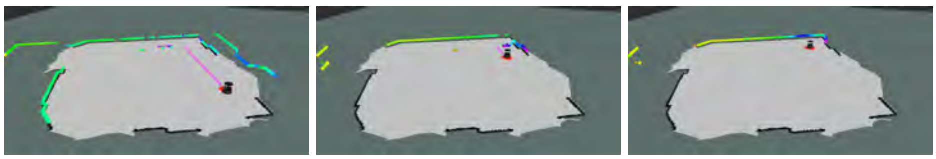


Fig. 5. Robot planning and executing a trajectory to reach a given target location in the map (visualization in *RosViz*). The thick green and blue contours matching map contours are laser scan points aligned with the map. The thinner pink line starting at the robot represents the planned trajectory. The red segments around the robot represent the robot location uncertainty.

Mapping. The map collection was carried out using a standard mapping node available in ROS[9]. It implements the well known SLAM algorithm from [5] and allows us to easily build a map from 2D laser scan data. The obtained 2D map allows the robot to locate itself at any time, as long as the layout of the mapped region does not change significantly.

The map is stored as a model of free vs occupied spaces in the area. The map obtained on the performance scenario area is shown in Fig. 4(b). White region in the map means free space, while dark gray means out of the map limits. In this representation we can see how the stored model enlarges the occupied/unreachable regions (purple shaded areas) around the obstacles and walls (black lines). This means that the robot is not allowed to get too close to the actual physical elements to avoid collisions. This feature is a common navigation safety feature that needs to be taken into account carefully when designing the choreography, since when using the map navigation features, the robot can not reach any position very close to any obstacle (including the dancers). We need to disable this for dynamic obstacles if we want to be able to have the robot move very close to the other dance team members, as it can be seen in many of the examples in this paper, such as Fig. 4(a).

Localization. The robot localizes itself on the available map using the 2D laser scan data. This method enhances the first attempt based only on odometry which

[9] http://wiki.ros.org/gmapping

is only useful for short periods of time. The system uses the available AMCL localization ROS node[10]. This localization approach is a well known probabilistic 2D localization method which implements an adaptive Monte Carlo localization approach using a particle filter to track the robot pose on a known map, as described in [16]. Fig. 5 shows some of the main ingredients of this approach. The robot uncertainty about its location depends on how well the algorithm is able to align the current 2D scan data with the known map of the environment.

3.2 Architecture of the System

The system used to control the robot in this project has two main components, to control the robot movements and to control the LEDs.

Motion module. The system to control the robot has been built using ROS and the available modules to control the TurtleBot2 platform [11]. There are two essential elements for our motion module: odometry estimation and localization on the scenario map. Figure 5 shows several snapshots of the robot executing a trajectory following a local planner to reach a goal. However, there are many situations where it is impossible to perform this planned movement towards a goal. This is because often the movement to be performed by the robot must be more precise in both position and time of execution, because it has to be synchronized with the music. In these cases, the robot has to execute movements purely based on its odometry.

A diagram of the main ROS nodes and topics used in our system is shown in Fig. 6. It is composed by several ROS modules which provide the robot with the capability of autonomous navigation (red on *move_base*) and localization (green ones *map_server*, *amcl*). The navigation is based on a reactive obstacle avoidance method and also incorporates a planner for computing the path to the goal assigned. The localization modules provides the location of the robot during the performance. It is based on a pre-built map and a particle filter which provides a localization. The blue module *figuras_node* is synchronized with the music, and it is in charge of the control of all the performance choreography (details on the specific trajectories generated are given in next section). It sends velocity commands to the low lever robot controller in order to perform a movement, and also sends goals in the global frame of the map to the navigation system.

LEDs module. We used 2 LED strips, each one containing 32 *RGB*-programmable LED components. We use the high-level API provided by the manufacturer [12] and focus on the colour and timing programming of each LED. The LEDs humanize the robot. One strip is used to emulate the robot eyes and other body parts. The second strip is just for aesthetic purposes. In addition to the aesthetic goals, these LEDs help the dancers to see in which modules the robot is running

[10] http://wiki.ros.org/amcl
[11] http://wiki.ros.org/indigo, http://wiki.ros.org/turtlebot_navigation/
[12] https://github.com/arvydas/blinkstick-python

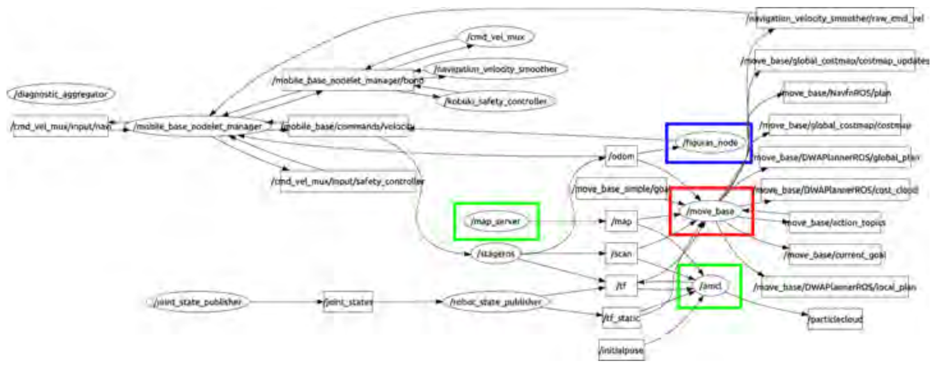


Fig. 6. Diagram of the main ROS nodes and topics used in our system.

at all times. This allows them to synchronize with the robot and between them using the LEDs as a reference. The LEDs module is run in parallel threads to the main process at all times, selecting different behaviours depending on the part of the choreography, such as moving forward or backward, stopped looking to the public or stopped waiting to synchronize with the timings. The LED stripe enables separated programming of the RGB color of each LED, therefore we could easily configure which LEDs correspond to the eyes, heart or head of the robot.

4 Experiments

This section details the set up used and the main results obtained from the experiments performed before and during the dance festival.

Set up. The final configuration and on-board sensors of the robotic platform used in all the experiments have been detailed in previous Sec. 2. About the particular details of the software platform, our system uses the Indigo ROS distribution[13] over Ubuntu 14.04 installed on the main on-board computer. The TurtleBot 2 platform has an embedded computer to control the low level motion modules of the robot, and an additional computer mounted on the robot plates where we the main software components, including ROS, are installed. Our platform is equipped with a Intel NUC 6i5SYH (Intel Core i5-6260U, 8GB DDR4).

Scenario. The performance of this project is executed on a large open space that belongs to a residence in the center of Art and Technology Etopia[14], shown in Fig. 7. Not all the space is mapped by the robot because a large part needs to be kept free for the public.

[13] http://wiki.ros.org/indigo

[14] https://www.zaragoza.es/ciudad/etopia/

Fig. 7. Panoramic view of live performance scenario. Red rectangle highlights where the actual performance happens. The public occupied the rest of the open space.

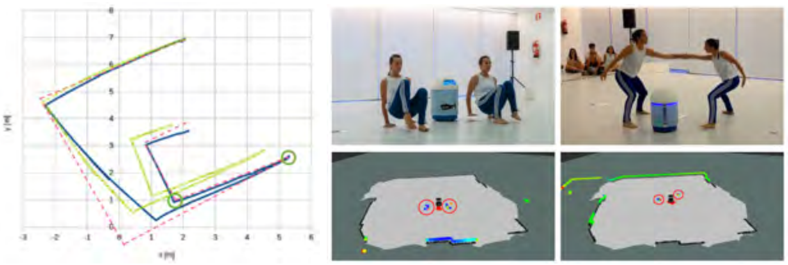
4.1 Choreography

The final choreography designed for the robot is represented in the diagrams from Fig. 9. The choreography consists of several figures whose execution took around 12 minutes, covering around 100m of different trajectories, with three different music/sound themes combined.

Robot movement types. As previously mentioned, the movements of the robot can be divided on two groups. Regarding *odometry-based* movements (blue trajectories on the figures), most of the specific trajectories need to follow certain geometric shapes and traverse exact distances in specific time intervals. In order to have tight control over this, many trajectories are executed using trajectory generation based purely on odometry estimation and verification. Regarding *map-goal-based* movements (green trajectories in the figures), they were used by the robot in several key steps along the choreography. These trajectories use the autonomous navigation module and the localization algorithm (AMCL ROS node) that locates the robot within the existing environment map. This requires varying waiting times at the end of this figures/movements, which are not ideal for the artists, but were necessary to avoid collisions due to inevitable accumulated odometry drift over time.

Robot movement error analysis. We executed the whole choreography trajectories in the simulation environment available for TurtleBot within ROS, but we observed that the execution times and odometry drifts were not equivalent, probably due to the varying weight, balance and set up of our platform compared to the original TurtleBot model incorporated in the simulator. Besides, we compared the ideal programmed trajectories, with the trajectory that would be followed using only odometry information, or our combined system of odometry and goal-based, and as it can be see in Fig. 8(a), the use of the goal-based movements is essential to be able to guarantee certain location of the robot at the times where the artist required high precision (e.g., because they were going to be moving very close to the robot). However, we should note that we can't establish map-goals all the time, because when the dancers are too close, as in the examples shown in Fig. 8(b), the map goal may be occluded and the robot will get stuck and lose the music synchronization.

Regarding *map-goal-based* movements, the acceptable error is directly configured in the ROS node configuration parameters, because they allow us to



(a) (b)

Fig. 8. (a) Choreography trajectory during first 3'. Ideal path (dashed pink), odometry estimation (light green) and final system (dark blue). Small circles are goals established on the map during this part. (b) Pairs of images from the public perspective and the robot monitoring tool (RViz viewer). The colored contours correspond to current 2D laser scan measurements. Red circles mark laser points corresponding to dancers.

establish the acceptance threshold to consider that the robot has reached a position. Regarding *odometry-based* movements, there is higher variability on the error values and significance. The accumulated error from the odometry was not significant when the accuracy of the trajectory shape and timing were more important for the artists than the particular location in the map where it is being performed (e.g., the circular trajectories in Fig. 9(h)). However, in other choreography figures a small error in odometry estimation means a collision with the walls (e.g., end of Fig. 9(a)) or with the dancers (e.g., beginning of Fig. 9(i)). When the robot traverses a narrow space near the dancers, it is very likely that the dancers accidentally push the robot and strongly affect the odometry. In these parts, the robot moves autonomously to a specific location in the map, rather than following a particular trajectory and speed.

4.2 Live Performance Results

As planned, the final demonstration of the choreography was performed on the *Trayectos festival*[15]. The festival performance consisted on two executions of the choreography, both of which were executed perfectly without any issues, in a full scenario, with more than 100 people in the public for each of them. Figure 8(b) shows several moments from the performance from the public perspective and from the robot monitoring system. Note how close the robot and the dancers move in some of the figures. Fig. 10 shows images of the performance. Note the difficulty to synchronize some of the designed choreography figures.

5 Conclusions

This paper has presented our experience and results integrating a low cost robotic platform on a dance performance. The main goal was to integrate the robot as

[15] Video available online: http://robots.unizar.es/data/videos/robot17Etopia.mp4

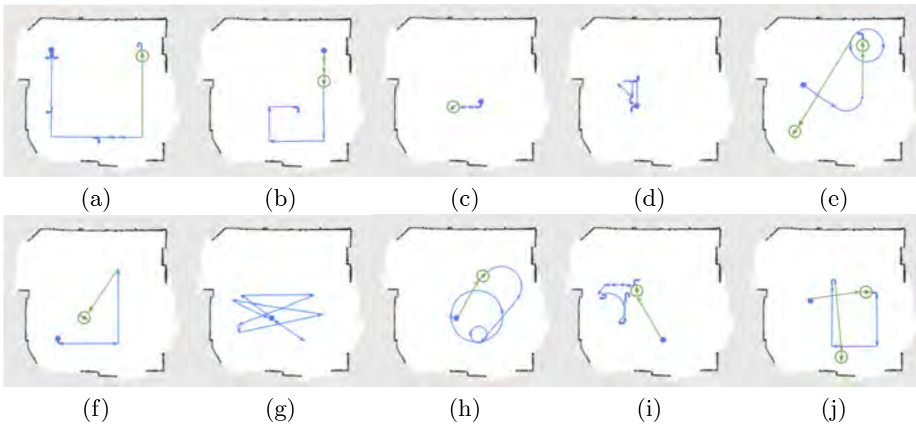


Fig. 9. Robot Choreography. Sorted diagrams (a)-(j) representing the movements performed by the robot. Blue: *odometry-based* movement. Green: *map-goal-based* movement. Thick blue dot in each diagram: starting point in that part. The blue arrows indicate rotations and directions of movement.

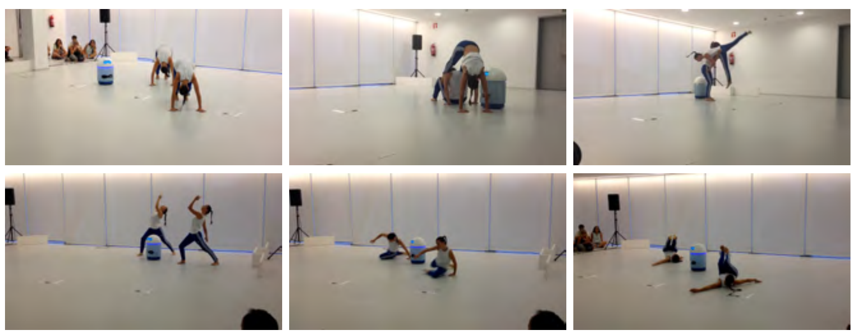


Fig. 10. Results from the live performance. Examples of synchronization figures where speed and length of the trajectory synchronization are essential.

part of the team in the dance performance. Therefore, the robot should have an accurate and repeatable choreography synchronized with the human dancers. The choreography was successfully designed and executed on the *Trayectos* festival as planned, thanks to a multidisciplinary team effort, between dance company, design and robotic engineers and researchers. Some of the most interesting lessons learned from this experiment have been the efforts required to bridge the gaps between the Art and Robotics worlds, starting from requirements and goals from each of the groups. From the artistic side, the aesthetics and accurate and repeatable movements were more critical than sophisticated localization algorithms. Besides, reactive behaviours were not feasible, since they can't ensure the strict time synchronization required. This initial experiment has open new paths to future collaborations. Integrating coordinated multi-robot teams and additional sensors and reactions are part of future work possibilities.

Acknowledgments. The authors would like to thank the support from the CESAR laboratories and its team, all the colleagues from the Robotics lab, as well as the work from all the people at *Trayectos* and *Tarde o Temprano*. This research has been partially funded the European Union (RT-ART - Grant agreement 645220), the Spanish Government (projects DPI2015-65962-R, DPI2015-69376-R) and Aragon regional government (Grupo DGA T04-FSE).

References

1. Aucouturier, J.J., Ogai, Y., Ikegami, T.: Making a robot dance to music using chaotic itinerancy in a network of fitzhugh-nagumo neurons. In: Neural information processing. pp. 647–656. Springer (2008)
2. Chan, M.T., Gorbet, R., Beesley, P., Kulič, D.: Curiosity-based learning algorithm for distributed interactive sculptural systems. In: IEEE/RSJ IROS. pp. 3435–3441 (2015)
3. Choi, T., Do, H., Kim, G., Kyung, J., Moon, J.Y.: An example of performing art with robot. In: Ubiquitous Robots and Ambient Intelligence. pp. 905–906 (2016)
4. Chung, C.C.J.: Integrated steam education through global robotics art festival (graf). In: Integrated STEM Education Conference. pp. 1–6. IEEE (2014)
5. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Trans. on Robotics 23(1), 34–46 (2007)
6. Herath, D., Kroos, C., et al.: Robots and Art: Exploring an Unlikely Symbiosis. Springer (2016)
7. Kapur, A., Darling, M., Diakopoulos, D., Murphy, J.W., Hochenbaum, J., Vallis, O., Bahn, C.: The machine orchestra: An ensemble of human laptop performers and robotic musical instruments. Computer Music Journal 35(4), 49–63 (2011)
8. Kosuge, K., Hayashi, T., Hirata, Y., Tobiyama, R.: Dance partner robot-ms dancer. In: IEEE/RSJ IROS. vol. 4, pp. 3459–3464 (2003)
9. Kudoh, S., Ogawara, K., Ruchanurucks, M., Ikeuchi, K.: Painting robot with multi-fingered hands and stereo vision. Robotics and Autonomous Systems 57(3), 279–288 (2009)
10. Martin, C., Hughes, J.: Robot dance: Edutainment or engaging learning. Proc. of the 23rd Psychology of Programming Interest Group PPIG 2011 (2011)
11. Nakata, T., Sato, T., Mori, T., Mizoguchi, H.: Expression of emotion and intention by robot body movement. In: Proc. of Int. Conf. on Autonomous Systems (1998)
12. Okamoto, T., Shiratori, T., Kudoh, S., Nakaoka, S., Ikeuchi, K.: Toward a dancing robot with listening capability: Keypose-based integration of lower-, middle-, and upper-body motions for varying music tempos. IEEE Trans. on Robotics 30(3), 771–778 (2014)
13. Peng, H., Zhou, C., Hu, H., Chao, F., Li, J.: Robotic dance in social roboticsa taxonomy. IEEE Trans. on Human-Machine Systems 45(3), 281–293 (2015)
14. Suzuki, R., Lee, J., Rudovic, O.: Nao-dance therapy for children with asd. In: ACM/IEEE Int. Conf. on Human-Robot Interaction. pp. 295–296 (2017)
15. Takeda, T., Kosuge, K., Hirata, Y.: Hmm-based dance step estimation for dance partner robot-ms dancer. In: IEEE/RSJ IROS. pp. 3245–3250 (2005)
16. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press