Social Group Motion in Robots

Francesco Zanlungo[1(✉)], Zeynep Yücel[1,2], Florent Ferreri[1], Jani Even[1], Luis Yoichi Morales Saiki[1,3], and Takayuki Kanda[1]

[1] ATR Labs, Kyoto, Japan
zanlungo@atr.jp
[2] Okayama University, Okayama, Japan
[3] Nagoya University, Nagoya, Japan

Abstract. Mobile social robots and (semi-)autonomous small size vehicles such as robotic wheelchairs need to understand and replicate pedestrian behaviour, in order to move safely in the crowd and to interact with, move along with and transport humans. A large amount of research about pedestrian behaviour has been undertaken by the crowd simulation community, but such results cannot be trivially adapted to robot applications. We discuss a simple but general recipe to apply an acceleration based pedestrian model ("Social Force Model") to mobile robots, and, as a specific example, we show how to replicate in a group of robots the behaviour of social pedestrian groups.

1 Introduction

Navigation of robots in pedestrian facilities presents features and requirements that are different from those of robot navigation in presence of only non-human obstacles, and this is particularly true for robots intended to interact with humans [18]. Such robots need indeed to predict human motion, and to move in a way that is perceived as safe and understandable by pedestrians, both for reasons of safety and to be able to accomplish their goals (e.g., approach humans in order to provide services [4,7,8], guidance [2,3,19], and the like). This is particularly true for robots that are supposed to assist humans, or just to behave as their companions [13], since these robots need to:

1. understand and predict the movement of pedestrians in the crowd (including the people they are assisting) in order to avoid colliding with them or hindering them,
2. move in a pedestrian-friendly way, i.e. in a way that other pedestrians may easily understand and perceive as safe,
3. move together with the people they are assisting, i.e. staying not only spatially close to them, but also in a position that is perceived as comfortable for interactions by the humans.

A specific instance of this problem is the realisation of a small-size vehicle for automatic or semi-automatic navigation in pedestrian areas (stations, shopping mall, sidewalks and the like), namely the realisation of a robotic wheelchair

A. Kheddar et al. (Eds.): ICSR 2017, LNAI 10652, pp. 474–484, 2017.
https://doi.org/10.1007/978-3-319-70022-9_47

[11,12]. Such a vehicle would, indeed, not only need to be able to safely avoid other pedestrians, and to be perceived as safe by them, but also to be able to move along with other pedestrians and similar vehicles to allow its passenger(s) to enjoy interaction and social communication.

A possible strategy to handle this problem is to reproduce in the robot pedestrian behaviour, i.e. to have the robot move just like the other people in the crowd, since this would not only reproduce in the robot the knowledge of human behaviour and the safe navigation skills that humans posses, but also provide the robot with a behavioural pattern to which other humans are used. It is obviously not clear that humans will perceive a robot as maximally safe and understandable if it behaves "as a pedestrian", but such an hypothesis is a reasonable starting point confirmed by preliminary investigation works [14,15,18,20]. Such an approach is definitely more simple and computationally economical than providing the robot with a full scale model of crowd behaviour, although the latter approach may be needed when more complex navigation tasks are involved [10].

For these reasons, researchers working in the field of human robot interaction with mobile robots often take advantage from pedestrian behaviour models developed in the field of pedestrian and crowd simulation [14,18]. It is nevertheless important to stress the fundamental differences in focus and scope of the two fields. While robotics researchers are usually interested in controlling a single (or at most a few) agent in the most reliable and safe way, crowd simulators are interested in reproducing the behaviour of a large number of agents, focusing more on the evolution of macroscopic observables such as crowd density than on the detailed behaviour of individuals. In crowd simulation, the motion of agents is often controlled by expressing their acceleration as a function of the position and velocity of neighbouring pedestrians (i.e. using a "Social Force Model" [6]), which is extremely efficient for fast simulation but needs careful consideration at the time of application on a mobile robot.

Being interested both in crowd simulation and its application to human robot interaction, we have developed algorithmic models of pedestrian behaviour that focus on a realistic reproduction of individual behaviour, with a particular interest in group behaviour [22–25], an aspect of crowd dynamics which is often neglected but, as stated above, extremely important for robots that may need to understand the idiosyncrasy of group behaviour and potentially reproduce it to move along with pedestrians [13]. In this work we show how our model (and similar acceleration based models) may be implemented in a non-holonomic mobile robot, taking into account its motor and sensory limitations.

2 Problem Formulation

The Social Force Model, originally introduced by Helbing et al. [6], is actually a class of models or a framework, in which the motion of the pedestrian is expressed through a second order differential equation. For the purpose of this work, we may not consider the physical forces deriving by body contact (since in the case of robotic applications such collisions have to be necessarily avoided) and the stochastic terms expressing the unpredictability of human behaviour, while we explicitly express group behaviour interaction. We may thus say that in a general Social Force Model for robot applications, the acceleration of a pedestrian is given by

$$\mathbf{a} = \mathbf{f}_{goal} + \mathbf{f}_{group} + \mathbf{f}_{ped} + \mathbf{f}_{obs}. \tag{1}$$

The first term, $\mathbf{f}_{goal}$, corresponds to a drag force that makes pedestrians walk at their preferred speed in their wanted direction. It is usually expressed as

$$\mathbf{f}_{goal} = \frac{1}{\tau}(v_p \mathbf{g} - \mathbf{v}) \tag{2}$$

where τ is a constant with dimension of time (characteristic time to recover one's velocity), v_p is the speed at which the pedestrian can comfortably walk (the pedestrian's *preferred speed*), and the unit vector $\mathbf{g}$ is the local goal of the pedestrian, possibly determined by a path planner.

The second term, $\mathbf{f}_{group}$, makes the pedestrian walk as a member of a group. In our works [22,23], we considered the need of pedestrians to keep their distance r[1] close to a maximum comfort one, r_0, while trying to minimise the angle θ that their gaze has to span between their walking goal $\mathbf{g}$ and their partners' gaze. We thus introduced the pair-wise potential between pedestrians as

$$U^\eta(r,\theta) = C_r\left(\frac{r}{r_0} + \frac{r_0}{r}\right) + C_\theta\left((1+\eta)\theta^2 + (1-\eta)(\theta - \mathrm{sign}(\theta)\pi)^2\right), \tag{3}$$

derived $\mathbf{f}_{group}$ as its gradient

$$\mathbf{f}_{group} = -\nabla U^\eta(\mathbf{r}), \tag{4}$$

and used Statistical Physics methods to analytically study its consequences and calibrate it on actual pedestrian behaviour. The parameter η makes groups walk slower than individuals outside groups, and causes 3 people groups to assume their typical V shaped formation. For details, refer to the original works.

The third term, $\mathbf{f}_{ped}$, makes pedestrians avoid collisions with other pedestrians (and possibly robots and other moving agents whose size is similar to the one of humans). In the original work by Helbing [6], it was expressed as a simple function of the pedestrian distance vector $\mathbf{r}$

$$\mathbf{f}_{ped} = A e^{-r/B}\frac{\mathbf{r}}{r}, \tag{5}$$

although following works have shown that this equation cannot reproduce general human behaviour, since it does not take in account the pedestrians' relative or absolute velocity. One possible solution has been introduced in [24] as

$$\mathbf{f}_{ped} = A\frac{v}{t}e^{-r(t)/B}\frac{\mathbf{r}(t)}{r(t)}, \tag{6}$$

[1] By distance between pedestrians, or more in general between moving agents, we usually mean the distance between their body centres.

where v is the pedestrian position, t is the time at which pedestrians will get closest if they do not change their velocities, and $r(t)$ is the distance they would attain at t, again in the case they do not change their velocities[2]. In [25] we show how to modify this term in order to introduce "social norms" such as a tendency to walk, avoid and overtake in specific directions, e.g. on the left or right sides[3]. Refer to the original works for further details.

Finally, the last term $\mathbf{f}_{obs}$ allows the pedestrians to avoid collisions with non-human obstacles, namely non-moving architectural elements. Most researchers state that $\mathbf{f}_{obs}$ assumes the same form as $\mathbf{f}_{ped}$, possibly with different parameter values, replacing the distance between pedestrians with the distance between the pedestrian and the obstacles. In general, since the obstacle may have a different size from a human, $\mathbf{f}_{obs}$ usually involves the computation of the distance between the pedestrian *and the closest point of the obstacle*, as explicitly stated in Helbing's original work [6].

We face three problems when applying Eq. 1 to a non-holonomic robot:

1. Equation 1 assumes obstacles to be polygonal lines (walls), to use the closest point of the wall in the computation of $\mathbf{f}_{obs}$. On the other hand, in robot systems obstacles are usually expressed as laser scan points (we assume that humans, robots and other moving agents of similar size can be automatically recognised by a tracking algorithm [1]).
2. Acceleration has to be converted to a (v, ω) command, v being the linear velocity and ω the angular velocity of the robot.
3. Robot velocity and acceleration limitations have to be taken in consideration. Specifically, we assume the robot to have a maximum acceleration a_{max}, a maximum speed v_{max}, and a maximum angular velocity ω_{max}.

3 Problem Solution

These problems may be solved using complex algorithms such as extracting polygonal walls from the scans and using a detailed model of the robot physical dynamics, but we provide here a simple recipe that may be applied to any robot, assuming only that the (possibly non-constant) time step of the robot control loop Δt is known. This recipe may be applied to any model in the Social Force Model framework, although the specific examples provided in figures, simulations and robot implementations are based on our previous works [23,24].

First of all, assuming that obstacles are expressed as scan points from a sensor, the force $\mathbf{f}_{obs}$, originally intended as sum over all segments in the polygon

$$\mathbf{f}_{obs} = \sum_{i \in N_{segments}} \mathbf{f}^i_{obs} \tag{7}$$

[2] Namely, we change the current distance with the future one "at the time of collision", computed using relative velocity information.

[3] Such social norms were not implemented in the simulator developed for this work.

is replaced by the average $\overline{\mathbf{f}_{obs}}$ force generated by all scans $i \in N_{scans}$ at a distance $r_i < R_{max}$, where R_{max} is an appropriate threshold[4].

$$\overline{\mathbf{f}_{obs}} = \frac{1}{N_{scans}} \sum_{i \in N_{scans}} \mathbf{f}^i_{obs} \tag{8}$$

R_{max} was introduced to prevent this average from yielding a very low value due to the sum of many small contributions. It may be applied without problems to the exponential forms of Eqs. 5 and 6. Nevertheless, such exponential forms may also be replaced with a trapezoidal force, in order to introduce the threshold in a continuous way and, assuming $\overline{r}$ has a value $\approx$0.25 m, i.e. similar to body radius, to make the model of [24] similar to the so called "Velocity Obstacles models" (interact if you predict a collision) [5][5], e.g. its magnitude given by

$$f(r) = \begin{cases} A & \text{if } r \leq \overline{r} \\ A\frac{R_{max}-r}{R_{max}-\overline{r}}, & \text{if } \overline{r} < r \leq R_{max} \\ 0 & \text{if } r > R_{max}. \end{cases}, \tag{9}$$

The acceleration is then computed as

$$\mathbf{a} = \mathbf{f}_{goal} + \mathbf{f}_{group} + \mathbf{f}_{ped} + \overline{\mathbf{f}_{obs}} \tag{10}$$

and, if its absolute value is larger than a_{max}, scaled to

$$\mathbf{a}' = \frac{a_{max}}{|a|}\mathbf{a} \tag{11}$$

If we ignored the motion limitation of the robot, its next velocity would be

$$\mathbf{v}_{next} = \mathbf{v}_{current} + \mathbf{a}'\Delta t. \tag{12}$$

In case

$$\mathbf{v}_{next} \cdot \mathbf{v}_{current} < 0, \tag{13}$$

the robot would be intended to invert its direction. If this is not possible without stopping and rotating until facing in the $\mathbf{v}_{next}$ direction, such a command should be passed to the robot, and the "social force" computed again after the rotation is completed. Otherwise the new linear velocity is set as

$$v = \min(|v_{next}|, v_{max}). \tag{14}$$

[4] In our implementation, R_{max} was fixed by an optimisation algorithm as $\approx$0.4 m. Since the term $\mathbf{f}_{obs}$ is not intended to provide the robot navigation around obstacles, which should be inferred by a proper navigation algorithm, but only "emergency" collision avoidance, a relatively small value of R_{max} is completely acceptable. Furthermore, since in the example we are using Eq. 6, such R_{max} corresponds to a threshold for the distance at the time of maximum approach, and not for the current distance, which may be considerably larger.

[5] When dealing with the collision avoidance force between two pedestrians, robots or other moving agents $\mathbf{f}_{ped}$, $\overline{r}$ should be replaced with the body radius, i.e. assuming the size of the robot is similar to the human one, $\approx$0.5 m.

We then compute the component of $\mathbf{a}'$ orthogonal to the current velocity

$$a_{\perp} = \frac{(\mathbf{v}_{current} \times \mathbf{a}')_z}{|\mathbf{v}_{current}|}, \tag{15}$$

$(\mathbf{v}_{current} \times \mathbf{a}')_z$ being the vector product component perpendicular to the plane. Using the circular motion formula, we have for the robot's angular velocity

$$\omega' = v/a_{\perp}. \tag{16}$$

In case $|\omega'|$ is larger than ω_{max}, we scale the angular velocity to

$$\omega = \frac{\omega_{max}}{|\omega'|}\omega', \tag{17}$$

otherwise we set $\omega = \omega'$.

The obtained values (v, ω) are a good approximation to the dynamics prescribed by the pedestrian model, compatible with the robot's motion limitation. They should nonetheless be checked using a safety system, e.g. [17], before being actuated in real mobile robots moving in uncontrolled environments.

4 Simulation Results

We implemented our system in a simulator (https://youtu.be/X1I0sJXzJBQ), using scan data from a real environment (a corridor on the third floor of ATR labs in Kyoto) and simulating a group of 3 robots moving in presence of 4 individual pedestrians. Both robots and pedestrians navigate to the end of the corridor using the path planning algorithm of [16], and, once they reach their goal, change direction and navigate to the other end. While the pedestrians are controlled directly by Eq. 1, the robots follow the solution of Sect. 3[6]. Robots have a maximum acceleration $a_{max} = 2\,\mathrm{m/s}^2$, a maximum linear velocity $v_{max} = 0.9\,\mathrm{m/s}$, and a maximum angular velocity $\omega_{max} = 1\,\mathrm{rad/s}$. Their "preferred speed" is $v_p = 0.8\,\mathrm{m/s}$. Sensory limitation is introduced by adding white noise to the position and velocity readings of the robots[7].

The robots assume a wide V formation [22] when walking in a wide portion of the corridor and close it to cross a narrow space (Fig. 1), open a gap to let one pedestrian pass and assume a line formation to let two pedestrians pass (Fig. 2). Previous works [9] obtained similar results using a rule system in pedestrian simulations, but here they are obtained as the result of a differential dynamical system while taking in account the locomotion limitation of robots.

[6] The difference between the pedestrians and the robots is clear when they change direction, since the former do it instantaneously, while the latter need to rotate.

[7] The used model introduces some minor changes in the equations of [23,24], which were introduced to maximise stability, in particular with respect to actual robot implementation. Furthermore, some parameters have been changed with respect to the original work by using a GA algorithm with the intent, again, of minimising robot collisions. Since the purpose of this work is not to present a maximally efficient robot system, but to provide a general recipe to convert a pedestrian model for robot use, we leave these details for an incoming extended technical paper.

Fig. 1. Left: group of robots moving in V formation. Right: group of robots crossing a narrow passage (obstacles in black, direction of motion shown by white section).

Fig. 2. Left: group of robots (red) letting a pedestrian (blue) pass. Right: group of robots letting one pedestrian pass while overtaking a slower pedestrian. (Color figure online)

5 Robot Implementation

We also proceeded, in a very preliminary way, to implement the group behaviour Social Force Model on a group of 3 robots. From a conceptual point of view, this consists basically in determining the robots' (v, ω) command through the Social Force Model of Sect. 3, that takes as an input the robot's own goal, position and velocity, plus the position of the other robots, the obstacle scan readings, and possibly the positions and velocities of moving obstacles such as pedestrians (although in our preliminary implementation we did not deal with the problem of pedestrian recognition and avoidance). The function implementing the (v, ω) command based on such information was then simply copied from the C++ code of the simulator of Sect. 4 into the robot control system code.

Nevertheless, implementing the system on actual robots presents a few technical challenges that are not present in the development of a simulator, namely how to organise the flow of information from the real world to the (v, ω) command module, and how to perform such command in safety on the robot. In this section we briefly illustrate the technical details of the implementation process.

We implemented our system on 3 icart-mini robots from the T-frog project http://t-frog.com/ using the Robot Operating System (ROS) framework. For reasons of safety and easiness of implementation, we first developed our system in a virtual environment, namely using the same real world scan readings that we used in the C++ simulator of Sect. 4, and performing the robot movement by updating the position of each robot in the virtual environment, and only in a second step we actually implemented it on the mobile robots navigating the actual corridor with real time scan readings. Since, nevertheless, the ROS code was the same in the virtual and actual robot implementation (the only difference being having noise in the real time scan readings, and actuating the robot motion in the real physical world), we will limit ourselves to describe the latter.

All robots used their own odometry, estimated by the robot driver, the reading of their frontal Laser Range Finder, and their copy of the environment map, to estimate their own position using a particle filter localisation algorithm.

Each robot has a server that publishes its own position in the map (estimated by the algorithm above) and its own linear and angular velocities (estimated by the robot driver), and listens (socket driver) to get the positions and velocities of the other robots, and its own navigation goal[8].

These inputs are all converted to the same frame (the robot's own frame), and passed to the (v, ω) command module, that computes the next linear and angular velocities which are then passed to the robot driver. Nevertheless, the robot motion is also subject to an higher priority safety limiter, that sends a stopping command if an obstacle is too close (according to the laser scans and the odometry). Furthermore, the robot is also subject to a joystick input, that has the highest priority (used by a human controller only in case of danger).

An overview figure showing the processing on board one of the robots, and a snapshot of the actual 3 robot navigating the corridor are shown in Fig. 3, while a video showing the robots navigating the corridor is available, along with a description of the simulation results, at https://youtu.be/5B71v_bjDGY.

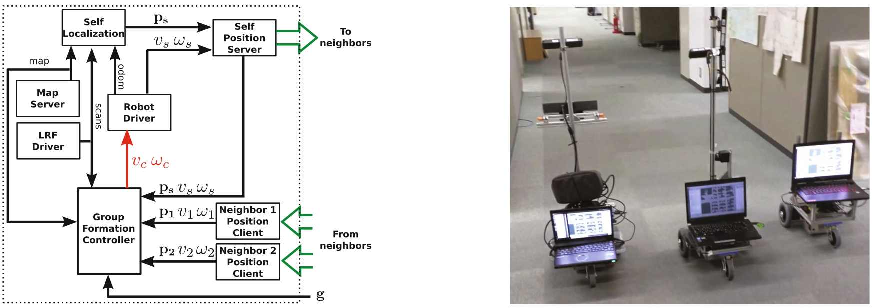

Fig. 3. Left: Graphical representation of the processing on board one of the robots. $\mathbf{p}$ stands for the robot position, v for the linear velocity and ω for the angular one. v_c and ω_c are the velocity commands as determined by the Social Force module, the subscript s refers to the robot itself while 1 and 2 to the neighbours. Green arrows represent socket communications between robots. Right: Group of 3 robots navigating according to the group model described in Sect. 3. (Color figure online)

[8] In this preliminary implementation, robots just navigated to the end of the corridor, and the goal was decided in advance.

6 Conclusion

We may expect that in the recent future many robots will need to be able to navigate inside pedestrians crowds, and thus to be able to understand and reproduce human pedestrian behaviour. This is particularly true for robots that are expected to interact with and assist humans (social robots), and for vehicles such as robotic wheelchairs. An understanding of group behaviour, in particular, may be extremely important. For all these reasons, it may be necessary to implement crowd simulation and pedestrian behaviour models in robotic systems, but the process may be non-trivial due to the differences in scope between the two fields. In this work we proposed a simple recipe to perform the transition from an acceleration based ("social force") pedestrian model to mobile robots, showing in particular how pedestrian social behaviour may be reproduced in a group of robots. We first implemented our system in a simulator, showing that the simulated robots reproduce complex and realistic group motion and collision avoidance while moving through a narrow and irregular corridor in presence of pedestrians, and then proceeded to a preliminary implementation in a system of 3 physical mobile robots. In order to actually deploy our system in a real envirnment we plan, as the subject of future work, to implement in our preliminary system actual tracking of other robots and pedestrians, and evaluate its safety in complex, changing, crowded and partially unknown environments [20,21].

Acknowledgements. This research is partially based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

1. Brščić, D., Kanda, T., Ikeda, T., Miyashita, T.: Person tracking in large public spaces using 3-D range sensors. IEEE Trans. Hum. Mach. Syst. **43**(6), 522–534 (2013)
2. Burgard, W., Cremers, A.B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S.: The interactive museum tour-guide robot. In: AAAI/IAAI, pp. 11–18 (1998)
3. Garrell, A., Sanfeliu, A.: Cooperative social robots to accompany groups of people. Int. J. Robot. Res. **31**(13), 1675–1701 (2012)
4. Gross, H.M., Boehme, H., Schroeter, C., Müller, S., König, A., Einhorn, E., Martin, C., Merten, M., Bley, A.: TOOMAS: interactive shopping guide robots in everyday use-final implementation and experiences from long-term field trials. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009, pp. 2005–2012. IEEE (2009)
5. Guy, S.J., Chhugani, J., Curtis, S., Dubey, P., Lin, M., Manocha, D.: Pledestrians: a least-effort approach to crowd simulation. In: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 119–128. Eurographics Association (2010)
6. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature **407**(6803), 487–490 (2000)

7. Huttenrauch, H., Eklundh, K.S.: Fetch-and-carry with CERO: observations from a long-term user study with a service robot. In: Proceedings of the 11th IEEE International Workshop on Robot and Human Interactive Communication, pp. 158–163. IEEE (2002)
8. Kanda, T., Shiomi, M., Miyashita, Z., Ishiguro, H., Hagita, N.: A communication robot in a shopping mall. IEEE Trans. Robot. **26**(5), 897–913 (2010)
9. Karamouzas, I., Overmars, M.: Simulating and evaluating the local behavior of small pedestrian groups. IEEE Trans. Visual. Comput. Graph. **18**(3), 394–406 (2012)
10. Kidokoro, H., Kanda, T., Bršcic, D., Shiomi, M.: Will I bother here?-a robot anticipating its influence on pedestrian walking comfort. In: 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 259–266. IEEE (2013)
11. Kuno, Y., Shimada, N., Shirai, Y.: Look where you're going [robotic wheelchair]. IEEE Robot. Autom. Mag. **10**(1), 26–34 (2003)
12. Morales, Y., Kallakuri, N., Shinozawa, K., Miyashita, T., Hagita, N.: Human-comfortable navigation for an autonomous robotic wheelchair. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2737–2743. IEEE (2013)
13. Murakami, R., Morales Saiki, L.Y., Satake, S., Kanda, T., Ishiguro, H.: Destination unknown: walking side-by-side without knowing the goal. In: Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction, pp. 471–478. ACM (2014)
14. Pacchierotti, E., Christensen, H.I., Jensfelt, P.: Evaluation of passing distance for social robots. In: The 15th IEEE International Symposium on Robot and Human Interactive Communication, ROMAN 2006, pp. 315–320. IEEE (2006)
15. Ratsamee, P., Mae, Y., Ohara, K., Kojima, M., Arai, T.: Social navigation model based on human intention analysis using face orientation. In: IROS 2013, pp. 1682–1687. IEEE (2013)
16. Schultz, M., Kretz, T., Fricke, H.: Solving the direction field for discrete agent motion. In: Bandini, S., Manzoni, S., Umeo, H., Vizzari, G. (eds.) ACRI 2010. LNCS, vol. 6350, pp. 489–495. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15979-4_52
17. Seder, M., Petrovic, I.: Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. In: 2007 IEEE International Conference on Robotics and Automation, pp. 1986–1991. IEEE (2007)
18. Shiomi, M., Zanlungo, F., Hayashi, K., Kanda, T.: Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model. Int. J. Soc. Robot. **6**(3), 443–455 (2014)
19. Thrun, S., Bennewitz, M., Burgard, W., Cremers, A.B., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., et al.: MINERVA: a second-generation museum tour-guide robot. In: Proceedings of 1999 IEEE International Conference on Robotics and Automation, vol. 3. IEEE (1999)
20. Truong, X.T., Ngo, T.D.: Dynamic social zone based mobile robot navigation for human comfortable safety in social environments. Int. J. Soc. Robot. **8**(5), 663–684 (2016)
21. Truong, X.T., Yoong, V.N., Ngo, T.D.: RGB-D and laser data fusion-based human detection and tracking for socially aware robot navigation framework. In: ROBIO 2015, pp. 608–613. IEEE (2015)
22. Zanlungo, F., Ikeda, T., Kanda, T.: Potential for the dynamics of pedestrians in a socially interacting group. Phys. Rev. E **89**(1), 012811 (2014)

23. Zanlungo, F., Kanda, T.: A mesoscopic model for the effect of density on pedestrian group dynamics. EPL (Europhys. Lett.) 111(3), 38007 (2015)
24. Zanlungo, F., Ikeda, T., Kanda, T.: Social force model with explicit collision prediction. EPL (Europhys. Lett.) 93(6), 68005 (2011)
25. Zanlungo, F., Ikeda, T., Kanda, T.: A microscopic "social norm" model to obtain realistic macroscopic velocity and density pedestrian distributions. PloS One **7**(12), e50720 (2012)