MARL-Ped: a Multi-Agent Reinforcement Learning Based Framework to Simulate Pedestrian Groups

Francisco Martinez-Gil[a,*], Miguel Lozano[a], Fernando Fernández[b]

[a] *Departament d'Informàtica, Escola Tècnica Superior d'Enginyeria (ETSE-UV), Universitat de València, Avda. de la Universidad s/n 46100 Burjassot. Valencia. Spain*
[b] *Computer Science Department, University Carlos III of Madrid, Avda. de la Universidad 30, 28911 Leganés. Madrid. Spain*

Abstract

Pedestrian simulation is complex because there are different levels of behavior modeling. At the lowest level, local interactions between agents occur; at the middle level, strategic and tactical behaviors appear like overtakings or route choices; and at the highest level path-planning is necessary. The agent-based pedestrian simulators either focus on a specific level (mainly in the lower one) or define strategies like the layered architectures to independently manage the different behavioral levels. In our Multi-Agent Reinforcement-Learning-based Pedestrian simulation framework (MARL-Ped) the situation is addressed as a whole. Each embodied agent uses a model-free Reinforcement Learning (RL) algorithm to learn autonomously to navigate in the virtual environment. The main goal of this work is to demonstrate empirically that MARL-Ped generates learned behaviors adapted to the level required by the pedestrian scenario. Three different experiments, described in the pedestrian modeling literature, are presented to test our approach: i) election of the shortest path *vs.* quickest path; ii) a crossing between two groups of pedestrians walking in opposite directions inside a narrow corridor; iii) two agents that move in opposite directions inside a maze. The results show that MARL-Ped solves the different problems, learning individual behaviors with characteristics of pedestrians (local control that produces adequate fundamental diagrams, route-choice capability, emergence of collective behaviors and path-planning). Besides, we compared our model with that of Helbing's social forces, a well-known model of pedestrians, showing similarities between the pedestrian dynamics generated by both approaches. These results demonstrate empirically that MARL-Ped generates variate plausible behaviors, producing human-like macroscopic pedestrian flow.

Key words: Route-choice, path-planning, Sarsa(λ)

*Corresponding author
Email addresses: Francisco.Martinez-Gil@uv.es (Francisco Martinez-Gil), Miguel.Lozano@uv.es (Miguel Lozano), ffernand@inf.uc3m.es (Fernando Fernández)

1. Introduction

In the current state of the art there are several pedestrian simulation approaches that focuses on steering the individuals (microscopic simulation) to generate both individual and group pedestrian behaviors. Microscopic pedestrians models consider the individual interactions and try to model the position and velocity of each pedestrian over the time. Between the most representative microscopic seminal models of pedestrians we have the cellular automata models [1], behavioral rule-based models [2], cognitive models [3], Helbing's social forces model [4] and psychological models [5]. In a microscopic simulator, the individuals are simulated as independent entities that interact with the others and with the environment, taking decisions to modify its dynamic state (including the calculation of the sum of a set of forces as a kind of decision). The decision-making process in the microscopic simulators follows a hierarchical scheme [6]: strategical, tactical and operational. The destinations and path planning are chosen at the strategical level, the route choice is performed at the tactical level and the instantaneous decisions to modify the kinematic state are taken at the operational level. Several microscopic simulators that focuses on the reproduction of the local interactions work only at the operational level [7].

A common problem in the microscopic models is the relationship between the individual behaviors and the group behavior. Traditionally, the rule-based systems [8, 4] are the most popular in this area to simulate local interactions. However, due to the complexity of the multi-agent colision avoidance, it is difficult to generate a lifelike group motion that follows the local rules [9]. Most agent-based models separate the local interactions from the necessary global path planning. To do this, there are two main approaches. One is to precompute or user-edit a path-planning map that is represented as a guidance field [9] or as a potential and velocity field [10]. Other consists on separating the local and global navigation problems in a layered model [11]. To make that split inside the agent model has the advantage that intelligent or psychological properties to the agents behaviors can be introduced [5, 12]. One indicator that this relationship is correctly resolved is that certain collective patterns appear when groups of pedestrians are under specific situations, as happens in the real world. Several collective behaviors have been described to appear in specific group situations like lane formations in corridors [13], faster-is-slower effect [14] and arch-like cloggings at bottlenecks [15, 13]. Social forces and its variants [4], agent-based models [16] and animal-based approaches [17], are microscopic models that have being successful in emerging collective pedestrian behaviors using different approaches. In pedestrian modeling, the capability to reproduce these phenomena, collective behavior or self-organization phenomena, is an indicator of the quality of the model.

In this work, a Multi-agent RL-based framework for pedestrians simulation (MARL-Ped) is evaluated in three different scenarios that are described in Section 5. Each scenario faces a different simulation problem. This framework constitutes a different approach to the existent microscopic simulators, that uses learning techniques to create an individual controller for the navigation

of each simulated pedestrian. The MARL-Ped framework offers the following benefits:

1. Behavior building instead of behavior modeling. The user does not have to specify guidance rules or other models to define the pedestrian's behavior. Only high level restrictions over the behavior of the agents are included in the framework as feedback signals in form of immediate rewards (i.e. to reach to the goal is good and the agent gets a positive reward; to go out of the borders is bad and then it gets a negative reward).
2. Real-time simulation. The decision-making module of each embodied agent (pedestrian) is calculated offline. In simulation time only the addition of the pre-calculated terms of a lineal function is necessary to get the correspondent best action.
3. It is capable of generating emergent collective behaviors.
4. Multi-level learned behaviors. The resulting learned behaviors control the velocity of the agent, which is a task of the operational level, but they are also capable of path-planning and route choice, which are tasks corresponding to the strategical and the tactical levels respectively.
5. Heterogeneous behaviors. The learned behaviors are different for each agent, providing variability in the simulation. This heterogeneity is intrinsic to the learned behaviors.

The aim of our work is not to provide a new pedestrian model (that implies the matching with real data) but to create plausible simulations of pedestrian groups (in terms of its adequacy to the pedestrian dynamics) to be used in virtual environments. In this animation context, agent-based pedestrian simulation is an active research field [18, 10] which considers simulations that can vary from small groups to crowds. Through the mentioned experiments we demonstrate that MARL-Ped is capable of generating realistic simulations of groups of pedestrians solving navigational problems at different levels (operational, tactical, strategical), handling the individual/group behaviors relationship problem mentioned before to produce the emergence of collective behaviors.

In order to show that the learned behaviors resemble pedestrians, we compare our results with similar scenarios defined in the Helbing's social forces pedestrian model. This well-known model in the pedestrian modeling field, has common characteristics with MARL-Ped: it is a microscopic model that also uses a driving force to get the desired velocity of the agent. The comparison is carried out by fundamental diagrams and density maps that are common tools used in the pedestrian dynamics analysis.

The rest of the paper has the following sections. In Section 2 we present the related work. In Section 3, some fundamentals of RL and an overview of the framework is described. Section 4 describe the modules of MARL-Ped. In Section 5, we describe the configuration of the scenarios. In Section 6 and Section 7 the results are discussed, and in Section 8 the conclusions and future work are exposed.

2. Related Work

From the point of view of the theoretical foundations, our work has similarities with Hoogendoorn's pedestrian route-choice model [19]. In this work the authors propose a Bellman-based optimization process to optimize an utility function designed as a weighted sum of route attributes. Using dynamic programming, a value function is calculated for the different spatial regions and used to find the pedestrian's route. In our approach, the utility function is substituted by an immediate reward function that values the interactions carried out by the agent inside its environment. The main advantage is the substitution of the utility function, which implies the assumption of a model of the environment, by a reward function that critics the consequences of the actions carried out by the agent. As in Hoogendoorn's model, our approach also reproduces route choice problems. From crowd simulation (computer animation), other works have used optimization processes to model pedestrian behaviors. Recently, the work [20] extends the principle of least effort [21] to model the motion of humans in a crowd using an optimization-based framework. The Still's Ph.D. thesis (where the Legion crowd simulator is introduced) also uses the least effort algorithm [22]. The work of Helbing et al. [23] suggests that in real pedestrians, a learning process exists to optimize the automatic response that minimize collisions and delays. In our approach, the value function which constitutes the decision-making task of each agent of MARL-Ped, is calculated using RL techniques [24], that also implies a Bellman-based optimization process.

In the last years, the use of RL in studies related with computer graphics, animation and simulation has increased. For instance, the work [25] uses RL to learn a policy that selects frames from a collection of motion capture data to animate and control avatars that represent boxers sparring. The work [26] uses RL techniques to learn a policy that assembles a motion stream from short motion fragments to animate characters in game envionments. A similar idea is developed in the work [27] to animate human-like characters with obstacle-avoidance choosing the adequate frames from a collection of motion capture data.

In our approach, we use RL with different purpose, because instead of learning a decision-maker to select adequate frames to create animations, we put the learning processes in the real protagonists of the simulation to create autonomous agents that move inside a physical environment to solve a multi-agent task.

3. Background and general overview

In this section we give an overview of the RL basic concepts used in this work and present our overall approach for MARL-Ped.

3.1. RL background

RL is a well-founded field of the machine learning area devoted to solve sequential decision-making problems. A RL problem can be modeled as a Markov Decision Process (MDP). A MDP is a 4-tuple constituted by a state space S, an action space A, a probabilistic transition function $P : S \times A \times S \rightarrow [0, 1]$ and a reward function $\rho : S \times A \times S \rightarrow \mathbb{R}$. The state signal s_t describes the environment at discrete time t. Assuming A is a discrete set, in a state, the decision process can select an action from the action space $a_t \in A$. The execution of the action in the environment changes the state to $s_{t+1} \in S$ following the probabilistic transition function $P(s_t, a_t, s_{t+1}) = Pr\{s_{t+1} = s' \mid s_t = s, a_t = a\}$ that is, the conditional probability of going to the state s' at time $t + 1$ when being at time t in the state s and performing the action a. Each decision is accompanied by an immediate scalar reward given by the reward function $r_{t+1} = \rho(s_t, a_t, s_{t+1})$ that represents the value of the decision taken in the state s_t. The goal of the process is to maximize at each time-step t the *expected discounted return* defined as:

$$R_t = E\{\sum_{j=0}^{\infty} \gamma^j r_{t+j+1}\} \tag{1}$$

where the $\gamma \in [0, 1[$ parameter is the *discount factor* and the expectation is taken over the probabilistic state transition P [28]. Note that the discounted return takes into account not only the immediate reward got at time t but also the future rewards. The discount factor weights the importance of the future rewards. The Action-value function (Q-function) $Q^\pi : S \times A \rightarrow \mathbb{R}$ is the expected return of a state-action pair given the policy π:

$$Q^\pi(s, a) = E\{R_t \mid s_t = s, a_t = a, \pi\} = E\{\sum_{j=0}^{\infty} \gamma^j r_{t+j+1} \mid s_t = s, a_t = a, \pi\} \tag{2}$$

Therefore, the goal of the learning algorithm is to find an optimal Q^* such as $Q^*(s, a) \geq Q^\pi(s, a)\ \forall s \in S, a \in A, \forall \pi$. The optimal policy $\pi^*(s)$ is automatically derived from Q^* as it is defined in Equation 3.

$$\pi^*(s) = \underset{a}{\operatorname{argmax}}\ Q^*(s, a) \tag{3}$$

If the transition probability function P is known, dynamic programming can be used to calculate Q^*. When it is not known, like in our problem domain, RL can be used.

3.2. MARL-Ped overview

MARL-Ped is a Multi-agent framework that has two kinds of agents: the *learning agents* and the *environment* agent. While the number of learning agents is defined by the experiment, there is only one environment agent. MARL-Ped has two working modes: the learning mode and the simulation mode. In the learning mode, a learning agent uses RL to learn a near-optimal value function

Q^*, able to control at each moment the velocity of the assigned virtual pedestrian. Once it has been learned, constitutes the core of the agent's decision-making module. In the simulation mode, the learning agents follow the near-optimal policy $\pi^*(s)$ derived from Q^* using the Equation 3. The environment agent works in the same way in both modes. It is in charge of the 3D virtual environment, where each learning agent is represented by an embodied virtual pedestrian. Each agent has been designed as an independent computational process which follows a distributed memory model of parallel architecture that uses the Message Passing Interface (MPI) programming model [29]. The communication takes place between each learning agent and the environment. Thus, there is not communication among the learning agents.

The dynamics of MARL-Ped is time-step based. At each time slot, t, all the learning agents interact with the environment following these steps:

1. Step 1: Each learning agent receives from the environment agent individual raw data that describe the current state, s_t, and a reward, r_{t-1}, that evaluates the previous decision making at step $t-1$. The reward value will be zero if the environment does not have information to judge the adequacy of the action.
2. Step 2: Each learning agent converts the received raw data into a generalized state space s_t.
3. Step 3: Each learning agent selects an action to be carried out. In learning mode, the state s_t and the reward r_{t-1} are used by the learning algorithm.
4. Step 4: The environment agent gets the actions of the learning agents and execute them. Each learning agent controls the behavior of a specific virtual pedestrian of the environment. The new actions modify the dynamics of the embodied virtual pedestrians. Then, the scene is simulated with the new dynamics during the rest of the time slot.

In simulation mode, the environment agent generates a file (divided in frames) with temporal information about positions and velocities of the embodied virtual pedestrians which constitutes the input for the graphics engine.

4. MARL-Ped framework description

In Figure 1, a functional diagram of MARL-Ped's agents is displayed. The modules have been enumerated with labels (M_i) to be more easily identified.

4.1. Learning agent's modules description

There are two abstract tasks in a learning agent: first the calculation of the generalized state space and, second, the decision-making process. When the learning mode is active, the decision-making task improves from scratch through the RL process. When the simulation mode is active, the decision-making task consists on following the learned policy π^*. The cost of calculating $\pi^*(s)$ using Equation 3 is constant, providing an efficient decision-making module appropiate for real-time simulations or interactive environments.

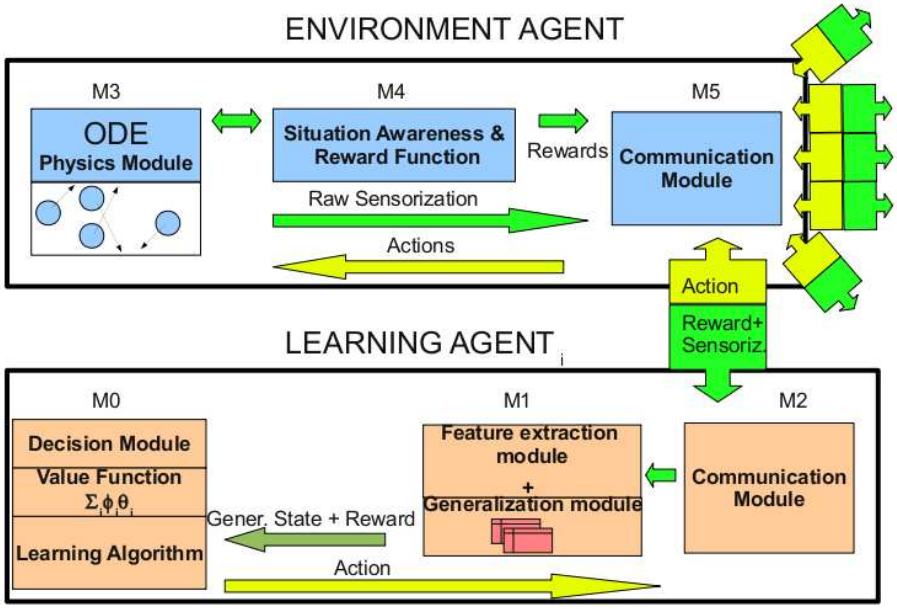


Figure 1: Functional schema of the MARL-Ped framework

Feature extraction and generalization modules (M_1)

Each agent receives from the environment raw information sensed by the assigned embodied virtual pedestrian. This information is converted into real features that describe the state of the agent.

The state space for each agent is modeled with the features showed in Figure 2. The chosen features provide local information about the own kinematic state of the agent, the relative kinematic state of the neighboring agents, and information about the position of the nearest static objects, like walls, respect to the agent. Similar features have been used previously in pedestrian models and they are considered as relevant for the kinematic description of the pedestrian [7] or to characterize the imminence of the collision [30]. It is important to note that this state space representation is fixed in the RL framework. In each specific experiment, a subset of these features is selected. For instance, in an environment without walls, the features related with the obstacles are disabled.

The features that describe the state are real valued, therefore a generalization process is needed to represent an usable value function. MARL-Ped allows exchanging the generalization module of its agents. For these experiments, we have chosen tile coding as the value function approximation method because it has been tested in our previous work giving good results [31]. Tile coding [24, 32] is a specific case of linear function approximation with binary, sparse features which is based on the Cerebellar Model Articulation Controller (CMAC) structure proposed by Albus [33]. It constitutes a specific case of the parameterized function approximators family where the functions are approximated with a lin-

Sag	Module of the velocity of the agent.
Av	Angle of the velocity vector relative to the reference line.
$Dgoal$	Distance to the goal.
$Srel_i$	Relative scalar velocity of the i-th nearest neighbor.
Dag_i	Distance to the i-th nearest neighbor.
Aag_i	Angle of the position of the i-th nearest neighbor relative to the reference line.
Dob_j	Distance to the j-th nearest static object (walls).
Aob_j	Angle of the position of the j-th nearest static object relative to the reference line.

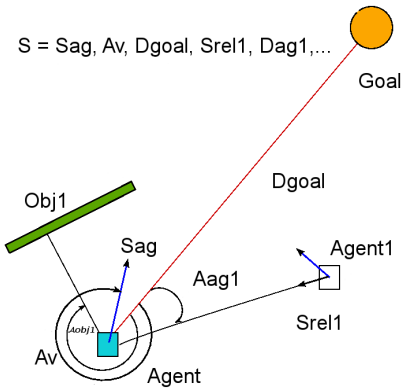


Figure 2: State space features. The reference line joins the agent with its goal.

ear combination of weighted binary-valued parameters. In tile coding, the space is divided exhaustively in partitions named tilings. Each tiling partitions all the space so there are as many partitions as tilings. Each element of a specific tiling is a tile and, given a point in the state space, there is only one active tile per tiling associated to this point. Given m tilings and k tiles per tiling, then $m \cdot k$ tiles exist. A binary vector $\vec{\phi}$ indicates the active tiles in each interaction at time t, and the vector $\vec{\theta}$ stores the value of the tiles. Therefore, for each tile i, $\phi_i(s)$ indicates if it is active (value 1) or not (value 0) for the state s. A weight stored in a table $\theta(i)$ indicates its value. The value function for each action Q^a and state s at time step t, is represented as a lineal combination as described in the Equation 4

$$Q_t^a(s) = \vec{\phi}^T \vec{\theta_t^a} = \sum_{i=1}^{m \cdot k} \theta_t^a(i)\phi_i(s) \tag{4}$$

where the super index T means the matrix transpose.

The code of a point of the state space is given by the binary features $\phi(i)$ that have value 1, remaining the rest with value 0. Therefore, in practice, the sum of Equation 4 is not over all tiles of all tilings since only one tile per tiling is active in the codification of a state. A critical problem of tile coding is that the memory requirement is exponential in the number of dimensions. In order to reduce memory requirements, a Hash function is used. By using it, a pseudo-random large tiling collapses into a much smaller set of tiles. It produces new tiles consisting of non-contiguous, disjoint regions randomly spread throughout the state space, but that still form an exhaustive tiling [24].

The learning module (M_0)

In our framework, the agents are independent learners that do not consider communication with the rest of the agents. The goal of the Multi-agent learning framework is to get a decision-based velocity controller for each agent. Specifically, each pedestrian learns a *policy* (i.e. a mapping between states to actions) that represents its navigational behavior and drives the agent towards the goal.

The core of the learning module is the RL algorithm. MARL-Ped uses Sarsa(λ) as the learning algorithm for the experiments described in this work. The algorithm Sarsa(λ) is a model-free policy iteration algorithm proposed by Rummery and Niranjan [34]. Sarsa(λ) belongs to the Temporal Difference algorithmic family (TD(λ)) characterized by reinforcing not only the state-action pair at the step t, but all the recent state-action pairs in a temporal window defined by the parameter λ. Sarsa(λ) starts with an arbitrary initialized Q-function and updates it at each step with the following update rule:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \, Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t) \right] \qquad (5)$$

where α is the learning rate and γ the discount factor.

The Sarsa(λ) algorithm is continuously estimating the value function Q using the policy π that is being learned. At the same time, the policy π is changing because the exploratory policy uses the new updated Q. In the learning process, the agent needs to explore the environment to find new better policies but also has to exploit the acquired knowledge to improve the expected return. In other words, the agent has to decide at each moment whether to act to gain new information (explore) or to act consistently with its past experience to maximize reward (exploit). The solution to this dilemma is to use an exploratory policy that balances the exploration with the exploitation inside the learning process. In the MARL-Ped framework, the user can define the exploratory policy for each experiment. For the experiments described in this work we have used a ϵ-greedy policy. In a ϵ-greedy policy, given a state, the policy selects the action with higher value with probability $1 - \epsilon$ and it selects a random action with probability ϵ. The parameter ϵ decreases with the time so as to the policy explores more at the beginning of the learning process and exploits more at the end, becoming a greedy policy at the final stages.

The modular architecture of MARL-Ped allows the selection of the techniques used both, in the generalization module and the learning module in different experiments (the reader can see our previous works with different configurations [35, 31]).

4.2. Environment agent's modules

The environment agent, that is unique, has several roles in the framework such as: sensing the parameters that define the raw state of each learning agent, to process the action instructions coming from the agents, to criticize the results of these actions through the reward signal and to manage the virtual environment.

Situation awareness and reward function module (M_4)

This module has a double function. First, it senses the virtual environment from the point of view of each embodied virtual pedestrian to collect the parameters that describe the raw state of the corresponding learning agent. These parameters are processed by the feature extraction module of the learning agent as explained in subsection 4.1. Second, the reward module is responsible for making a judgment of the suitability of the action selected at time t for each learning agent. The reward module defines the behavior of the agent and uses the delayed rewards in the calculation of the return (see Equation 1). This implies that the immediate reward signal is only part of the update term which takes also into account the future rewards that are consequence of the action. The reward function (which provides the immediate rewards) is designed by the user and it is not neccesary to model all the possible cases. When there is no information about the adequacy of an action taken in a specific state, a value of 0 is returned as the immediate reward.

Physics module and actions definition (M_3)

The physics module of MARL-Ped is a calibrated version of the Open Dynamic Engine (ODE) physics software library implemented by Russell L. Smith. The embodied agents have been modeled in ODE as spheres with radius 0.3 m and 50 Kg that agrees the mean wideness and weight of the human body. The friction forces among the agents and with the floor and walls have been specified in ODE with values of real pedestrians [31].

The dynamic model of the agents can be included inside the self-driven many-particle systems. In these systems, the particles have an internal energy reservoir that feeds an individual driving force [36]. Examples of these systems are animal herds, flocks of birds or traffic. In self-driven particles, the forces that actuate in an agent can be divided in two groups: external forces $\vec{F_j}$ that are generated by friction, collisions, pushing, etc. and an internal force $\vec{F^i_{driv}}$ to generate the agent's behavioral driving motion which is the objective of the learning process. The dynamic state of an agent i is described by Equation 6.

$$m_i\vec{a} = \vec{F^i_{driv}} - \mu_f|\vec{N}|\vec{u_v} + \vec{F_c} + \vec{F_{fr}} \tag{6}$$

Where $\vec{F_c}$ is the collision force, $\vec{F_{fr}}$ is the friction between the two objects and the second term is the friction force with the floor where μ_f is the friction coefficient, $\vec{u_v}$ is the unitary velocity vector and $\vec{N}$ is the normal force.

An agent's action can be understood as a behavioral force that the agent applies to itself to modify its velocity vector as real pedestrians do. The variation of this velocity vector has been used to control the trajectories in other pedestrian models [30] and it is carried out with two types of actions that actuate simultaneously. One type varies the speed; the other varies the direction. The agent has to choose a pair of actions (one of each type) in its decision. In terms of physics, each action pair is understood as the parameterization of the physic impulse $\vec{I} = m(\vec{v_{t2}} - \vec{v_{t1}})$ of the behavioral force $\vec{F^i_{driv}}$ from time t_1 to

t_2. There are eight different variation ratios in addition to the 'no operation' option for both, the speed and the direction, resulting in 81 possible combined actions.

5. Description of the simulated scenarios

In this section, the scenarios of the experiments are introduced. These scenarios model common situations for real pedestrians in urban environments.

The shortest path vs *the quickest path scenario*

In the problem of the shortest path versus the quickest path, an agent has to choose between two exits to reach to the goal. One exit is near the goal and the other is situated farther from it. If the number of agents is large, a jam is generated in front of the exit next to the goal. But other alternative path is available that detours this group using the other exit. Assuming that the extra effort to perform the detour is small, part of the agents of the borders of the jam can decide to follow the detour instead of keeping waiting inefficiently. This problem happens in different situations in real life (for example when pedestrians hustling through a station hall as they are late for a train) and it differentiates the pedestrian dynamics from other vehicle dynamics [37]. The pedestrian models that calculate the direction of the desired velocity of the agents using the shortest path to the destination (like Helbing's social forces [1]) or using the gradient of a potential field are not capable of reproducing the decision dilemma that generates this effect [38]. The layout of the environment in this experiment is a wall with two exits that divides a space in two areas. The agents are placed in the first area with dimensions 18 m wide and 6 m depth, and the goal to reach is placed in front of one exit in the second area. The exits are 1 m wide, which allow be traversed by only one agent at the same time. The two exits are separated a distance of 6 m. The distances from the nearest exit and from the farthest exit to the goal are 1.5 m and 6.2 m respectively (see Figure 8).

The crossing inside a corridor scenario

In the crossing problem, two groups of pedestrians walk in opposite directions inside a narrow corridor. The lane formation is an emergent collective behavior that appears in real situations where the pedestrians' movements are constrained by borders (real or not) like in urban sidewalks or corridors [13]. In everyday conditions, real pedestrians with opposite directions of motion do not equally distribute over the cross section of a congested walkway. On the contrary, pedestrians organize into lanes of uniform walking direction. This phenomenon maximizes the averaged velocity in the desired direction of motion, reducing the number of encounters with oppositely moving pedestrians.

[1] The Social Forces model has developed many variants since it was introduced originally by Helbing et al. We refer in this paragraph to its first conception.

The work [39] proves that lane formations in opposite groups of pedestrians can be derived from optimal states associated with minimal interaction and minimal dissipation. In our approach, the optimization process intrinsically associated to the RL algorithms will cause the emergence of the lanes. In our experiment, the dimensions of the corridor are 20 m long and 2 m wide (see Figure 11).

The maze scenario

Real urban pedestrians find mazes in several common situations. In a congested avenue, the vehicles create mazes that pedestrians have to solve to cross the pavement. In certain railway crossings, the pedestrians find fences building mazes that require them to approach the crossing turning left and right in order to force the visual detection of the presence of a train. To leave the maze, the virtual embodied agent needs to plan the trajectory. A greedy strategy, like that in which the agent always choose the straight path, can leave him stuck in a local minimum. Path-planning is a necessary task in crowd simulation commonly placed in the high level agent's behavior model [5] or in the environment [10]. In our approach, the path planning is a consequence of the RL dynamics and is intrinsic to the learned controller. The exploration policy assumes intrinsically the task of planning when forces the agent to explore different actions in a given state. In this experiment, the virtual environment is a square of 4×4 m where three walls have been situated adequately to create a maze. Two agents stay at the beginning of the episode in two adjacent corners of the square and their goals are placed in the diagonally opposite corner. The agents move across the maze in opposite directions (see Figure 12).

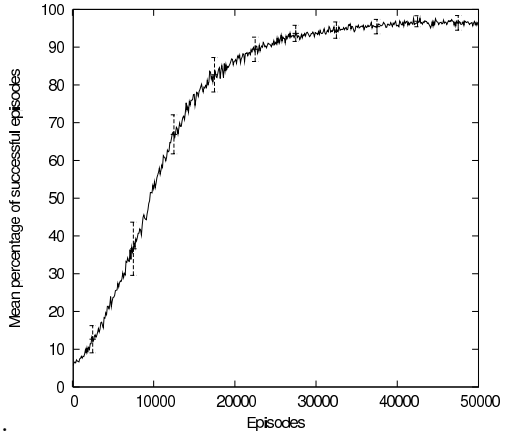
6. Learning results

In this section, the configuration as well as the performance reached by each learning process is described. There is not a fixed pattern to define the configuration of parameters and the strategies to be used in the learning process because each scenario has its own challenges that have to be addressed specifically.

6.1. Shortest path vs Quickest path

At the right side of the Figure 3, the main parameters of the learning process for this scenario are specified. The learning phase has 23 agents, that is a sufficient number to produce the bottleneck in front of the nearest exit to the goal. The learning process has been empirically fixed to a duration of 50000 episodes.

The left side of the Figure 3 shows the curve of the mean percentage of success of an agent. An agent solves successfully an episode when he/she reaches to the goal, independently of the chosen path. Each point in this curve represents the mean percentage of success for the last 100 episodes. The final asymptotic region of the curve (between the 30000 and the 50000 episodes), with a percentage of success of more that 90%, indicates that the agent has learned to solve the problem.



Number of learning agents	23
Number of episodes	50000
Parameter λ of Sarsa Algorithm	0.9
Learning rate	0.001
Influence of future rewards (γ)	0.9
Initial value of Epsilon (ϵ_0)	0.9
Number of tilings in Tilecoding	128
Reward when leaving the valid area	-10
Reward when the goal is reached	+100

Figure 3: Learning process for the Shortest path *vs.* Quickest path experiment. Left: Mean percentage of successful individual episodes. The curve shows the mean and standard deviation for the learning curves of 23 agents. Right: Values of the main parameters of each learning process.

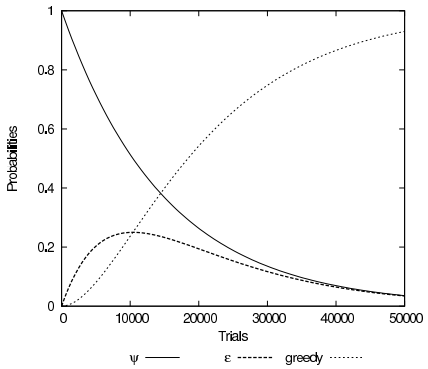
6.2. Crossing in a corridor

The crossing scenario is a problem of spatial organization in which anticipatory maneuvers of the pedestrians in each group can play an important role in solving the problem. A way to facilitate the search for a solution is to give useful information during the learning process. In the model-free RL framework there are several techniques to introduce information. We have used Policy Reuse [40], a transfer of knowledge method which uses an existent policy π_0 as a probabilistic bias for the exploring operation. In each decision, the learner chooses among the exploitation of the ongoing learned policy, the exploration of random unexplored actions and the exploitation of this existent policy. In our case, the existent policy π_0 consist on using actions that moves the agent towards the right side of the corridor. This policy is inserted inside the exploratory task of the algorithm and used with a probability ψ. The exploration-exploitation trade-off with Policy Reuse used in this work is defined in Equation 7. In Figure 4, the probability curves for the Policy-Reuse used are displayed. Both, the probability of using the existent policy and the probability of exploring random actions decreased with the number of episodes whereas the exploitation of the learned policy increased.

$$\begin{cases} \psi & \text{choose the } \pi_0 \text{ policy} \\ (1-\psi)\epsilon & \text{choose an aleatory action} \\ (1-\psi)(1-\epsilon) & \text{choose the greedy policy} \end{cases} \quad (7)$$

The specific values of the learning parameters for the crossing experiment are displayed in Figure 4. The averaged percentage of successful episodes for the 8

Number of learning agents	8
Number of episodes	50000
Parameter λ of Sarsa Algorithm	0.9
Learning rate	0.004
Influence of future rewards (γ)	0.9
Initial value of Epsilon (ϵ_0)	1.0
Number of tilings in Tile coding	64
Reward when the goal is reached	+100



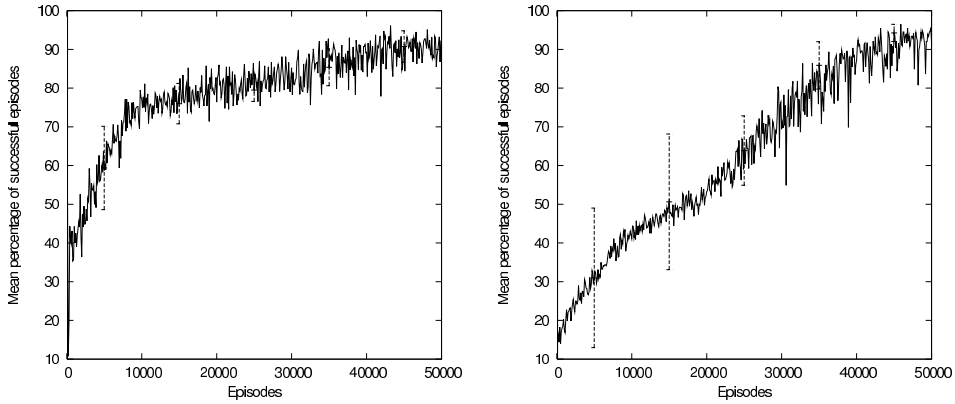
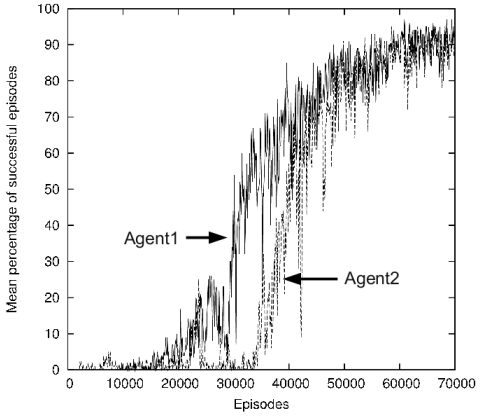


Figure 4: Crossing experiment. Left and Up: Values for the main parameters of the learning processes. Right and Up: Probability functions for the Policy-Reuse transfer method. Down: Performance of the learning processes. Left: Mean percentage of successful individual episodes for the experiment using Policy-Reuse. Right: Mean percentage of successful individual episodes for the experiment without Policy-Reuse. The means are calculated from the data of the 8 agents.

agents with and without using Policy-Reuse is also shown in this Figure 4. The use of Policy-Reuse has a positive effect in the learning processes as the agents learn faster. For example, at the episode 10000, the percentage of success with Policy-Reuse is about 75%, whereas in the same episode without Policy-Reuse is about 40%.

6.3. Pedestrians in a maze

The right side of the Figure 5 shows the table with the values of the parameters that configure the experiment. At the left side of the Figure 5 it is represented the percentage of successful episodes for each agent. The mean curve is not calculated in order to show how each agent learns. As it is shown, the agents do not learn at similar rhythm along the process. This asynchrony derives from the fact that the goal is discovered in different moments by each



Number of learning agents	2
Number of episodes	70000
Parameter λ of Sarsa Algorithm	0.9
Learning rate	0.007
Influence of future rewards (γ)	0.9
Initial value of Epsilon (ϵ_0)	0.3
Number of tilings in Tilecoding	32
Reward when leaving the valid area	-5.0
Reward when the goal is reached	+100

Figure 5: The maze scenario. Right: Values for the main parameters of the learning processes. Left: Individual curves of the percentage of successful episodes for each agent.

agent. As each agent perceives the other agent as part of the environment, a fast improvement in the policy of an agent creates a non-stationary environment for the other agent, that has more difficulty to learn. A careful adjustment of the learning rate (the α parameter) alleviates the problem. Despite this fact, the final values of success at 50000 episodes for both agents are about 90%, which indicates that the problem has been solved.

7. Simulation results

In this section, we highlight the performance of MARL-Ped on the described scenarios. First we introduce the tools used to analyze different aspects of the performance. Then, we present the results and illustrate them with videos that can be seen at the URL http::/www.uv.es/agentes/RL. These videos have been recorded and visualized in real time using Unity 3D Engine for the videos with 3D virtual environments.

7.1. Methodology

In the simulation phase, we used several common tools in the area of pedestrian dynamics to analyze the resultant behaviors: the fundamental diagrams and the density maps.

The fundamental diagram [41] describes the empirical relation between density and flow of pedestrians. It is a basic tool to design facilities like sidewalks, corridors or halls in public architectures [42] and it is a primary test of whether a pedestrian model is suitable for the description of pedestrian streams [43].

The form of the fundamental diagram used in this work (average velocity respect of the density) reveals the behavior of the velocity of the group in different situations (from free to congestion) in a specific point. For the calculation of the fundamental diagrams, we follow the specific formulation described in the works [44, 42]. Specifically, the local density is obtained by averaging over a circular region of radius R. The local density at the place $\vec{r} = (x, y)$ at time t is measured by

$$\rho(\vec{r}, t) = \sum_j f(\vec{r_j}(t) - \vec{r}), \tag{8}$$

where $\vec{r_j}(t)$ are the positions of the pedestrians j in the surrounding of $\vec{r}$ and

$$f(\vec{r_j}(t) - \vec{r}) = \frac{1}{\pi R^2} exp[-||\vec{r_j} - \vec{r}||^2/R^2] \tag{9}$$

is a Gaussian, distance-dependent, weight function. The local speeds have been defined via the weighted average

$$\vec{V}(\vec{r}, t) = \frac{\sum_j \vec{v_j} f(\vec{r_j}(t) - \vec{r})}{\sum_j f(\vec{r_j}(t) - \vec{r})}. \tag{10}$$

Secondly, we use a density map as it is a histogram that shows the occupancy of the physic space. The space is divided in tiles and the number of times that the tile is occupied along the time is counted. The density map reveals the patterns of movement present in the navigational problems (paths, cloggings, etc.) showing the zones frequently occupied by the pedestrians.

7.2. Shortest path vs. *quickest path*

In Figure 6 the density map created by MARL-Ped is displayed. The two flows of pedestrians that go through the exits towards the goal are visible. Note in the perspective view of the map, the clogging created around the exit of the shortest path (similar to a mountain with two picks) and the flow that deviates towards the quickest path represented by the shaded area between the values of Z=-4 and Z=2. The flat areas at X=0 corresponds to the walls (where the occupancy of the space is not possible). The flow of agents that use the quickest exit surrounds the longest flat area. In the side view, the ridge that connects the two heaps that represent the exits, also shows the trace of the agents that use the quickest path.

In Figure 7, the fundamental diagrams for Helbing's model and our MARL-Ped corresponding to a clogging in front of an exit, are displayed. The data of the Helbing's curve comes from the experiment of a closed room with an exit described in the work of Helbing et al [45] with the code available at the URL://http.pedsim.elte.hu. Although the set up of the experiment is not the same, the measure points for both diagrams are placed in front of the exit in which the clogging is created (in Helbing's experiment the exit is unique, in our experiment it corresponds with the shortest path exit). The diagrams compared show common characteristics: first a decreasing shape of the curves when the

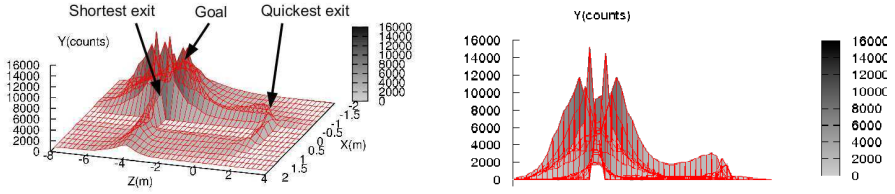


Figure 6: Density map for the shortest path vs quickest path experiment carried out with 23 agents. The data are from 100 simulations. Left: Perspective view. The shortest and quickest paths are visible. The exits are situated at Z=-4 and Z = 1 in the graphic (the coordinates do not correspond to real distances). The clogging in front of the shortest path exit corresponds with the area with highest densities. Right: Side view of the density map showing the two heaps corresponding to the two exits.

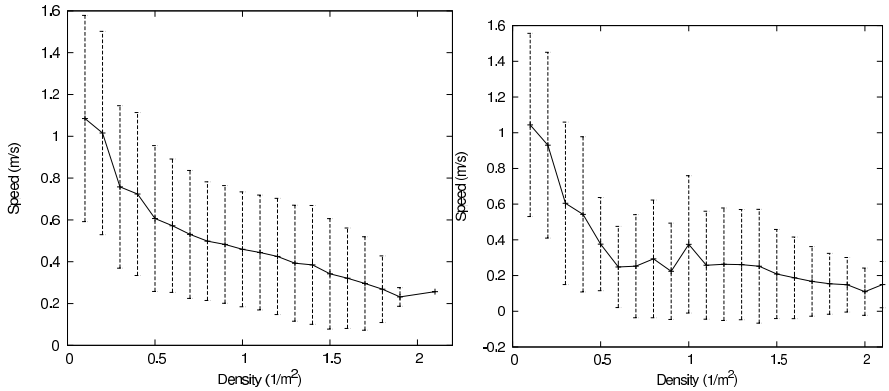


Figure 7: Fundamental diagrams for a measure point in a clogging in front of an exit with 23 agents. Left: Shortest vs Quickest experiment measured in front of the quickest path exit. Right: Helbing's model.

density increases, which is an important characteristic of the pedestrian dynamics. Second, the standard deviations of both curves show a decrease of the speed at high densities which is coherent with a state of congestion. The curves shapes are similar indicating comparable dynamics.

Table 1 summarizes the different performance measures carried out with 1000 episodes. In this table, individual average performance percentage (percentage of times that an agent reaches to the goal) and collective average performance percentage (average percentage of episodes in which all agents reach to the goal) are shown. While the individual percentage gives a measure of the quality of the individual learned behaviors, the collective percentage indicates the percentage of valid simulations (in which all agents reach the goal). The figures of the table show that more than 98% of the agents solve the problem and the 72.3% of the simulations end with all the agents finding the goal. In addition the 17.4% of the agents were able to find the alternative route to the shortest path.

In Figure 8, a temporal sequence of a simulation with 23 pedestrians is

Individual averaged performance percentage	Collective averaged performance percentage	Averaged outputs through the shortest path (from a total of 23 agents)	Average outputs through the quickest path (from a total of 23 agents)
98.6 ± 0.6	72.3 ± 4.5	17.0 ± 2.0 (73.9%)	4.0 ± 1.6 (17.4%)

Table 1: Performance analysis in simulation for the shortest path *vs* quickest path experiment. The individual averaged performance percentage counts the times (of 100 episodes) that an agent has reached to the goal (independently of the path). The collective averaged performance measures the number of times (of 100 episodes) in which all the agents arrived at the goal (independently of the path). The figures are averages of 10 experiments. For the averaged outputs, a set of 1000 episodes have been used.

displayed. The high density in front of the exit corresponding to the shortest path, generates a detour of some peripheral pedestrians towards the quickest path. Only the agents situated in the left side of the clogging selects the detour, as it is expected. Note in the images B and C of the sequence how two agents situated in the left border of the clogging use their quickest path selecting the farthest exit from the goal. When the clogging disappears, the agents choose again the shortest path. This situation can be seen with the pair of agents at the right of the shortest path exit in the image D of the sequence. Note how they disappear in the image E indicating that they use this exit to reach to the goal.

7.3. Crossing in a corridor

In this scenario we compare the experiments using MARL-Ped (with and without Policy-Reuse) and using the Helbing's model. In Figure 9 the density maps are displayed. The density map generated by MARL-Ped with Policy-Reuse shows two high density areas that correspond to the lanes. As can be seen in the first row of the Figure 9, the lanes appear near the walls of the corridor represented in the density map by two flat zones at both sides. On the contrary, the center of the corridor has a low occupancy. In the second row the map for the experiment without Policy-Reuse is displayed. The lanes also appear at both sides of the corridor, near the flat zones that indicate the presence of walls. The occupancy of the center is higher than that obtained in the previous experiment. Besides, the occupancy of the sides is not as clear as the previous mentioned experiment. These facts are indications that the agents do not create the lanes with the same anticipation and decision than in the experiment with Policy-Reuse (the agents use more often the central area of the corridor). The results of Table 2 also supports this difference between the two experiments as it is mentioned below. The third row corresponds to the Helbing experiment. The density map reveals that the space occupancy is different. The two groups of agents try to get the center of the corridor instead of deviating towards the walls as in the MARL-Ped results. It is an expected behavior because the presence of the walls generate repulsive forces that deviate the agents towards the center. When the individuals of both groups meet at half

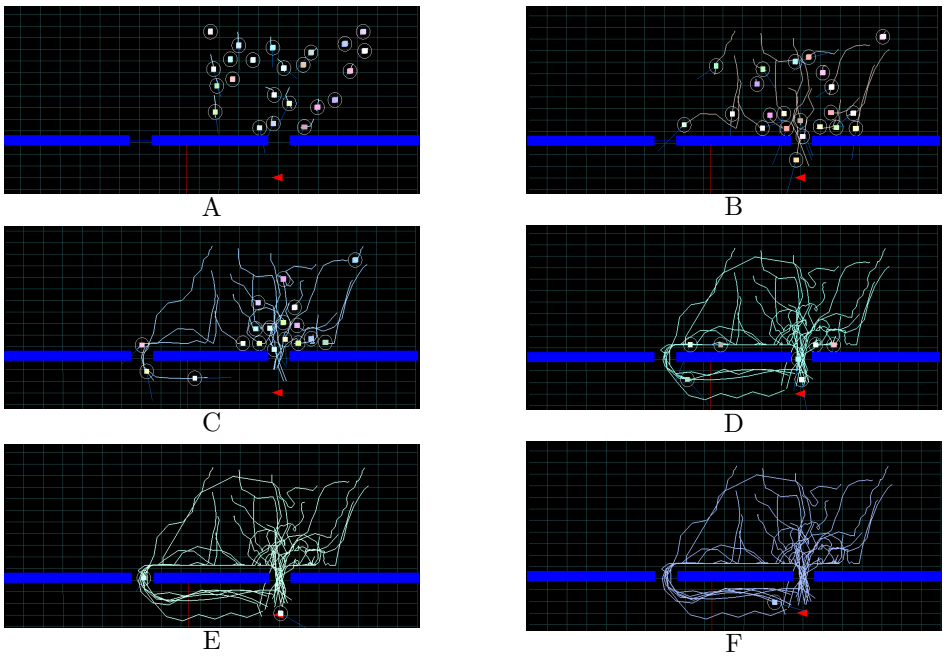


Figure 8: Sequence of stills from a simulation in the shortest *vs.* quickest experiment. The lines show the trajectories of the agents. The temporal sequence is sorted alphabetically.

distance of the goals, the repulsive forces between the agents allows the agents to approximate to the walls to solve the crossing. Therefore, a big occupancy is visualized in the map in the center of the corridor where the two groups of agents meet. The presence of lines in the Helbing's model is shaded by the central interaction zone.

The Figure 10 shows two fundamental diagrams corresponding to the MARL-Ped framework crossing experiment using Policy-Reuse (left column) and Helbing's crossing experiment (right column). The measure point has been chosen in the center of the corridor with a radius that covers the whole width of the corridor in both experiments. The main difference is the range of densities. In the Helbing's diagram this range cuts with a density of 0.9. This indicates that the crossings do not create congested situations. In the MARL-Ped experiment higher densities than the Helbing's experiment indicate more congested situations. On the other hand, the main characteristics of the diagrams described in the previous experiment also appear in this one, revealing similar pedestrian dynamics in both models.

In Table 2, a performance analysis of the crossing experiment with MARL-Ped is displayed. The meanings of the averaged individual percentage of success and the averaged collective percentages have been explained in the previous experiment. The figures of the performance show the important influence of Policy-Reuse in the collective performance, that is, in the rate of right episodes

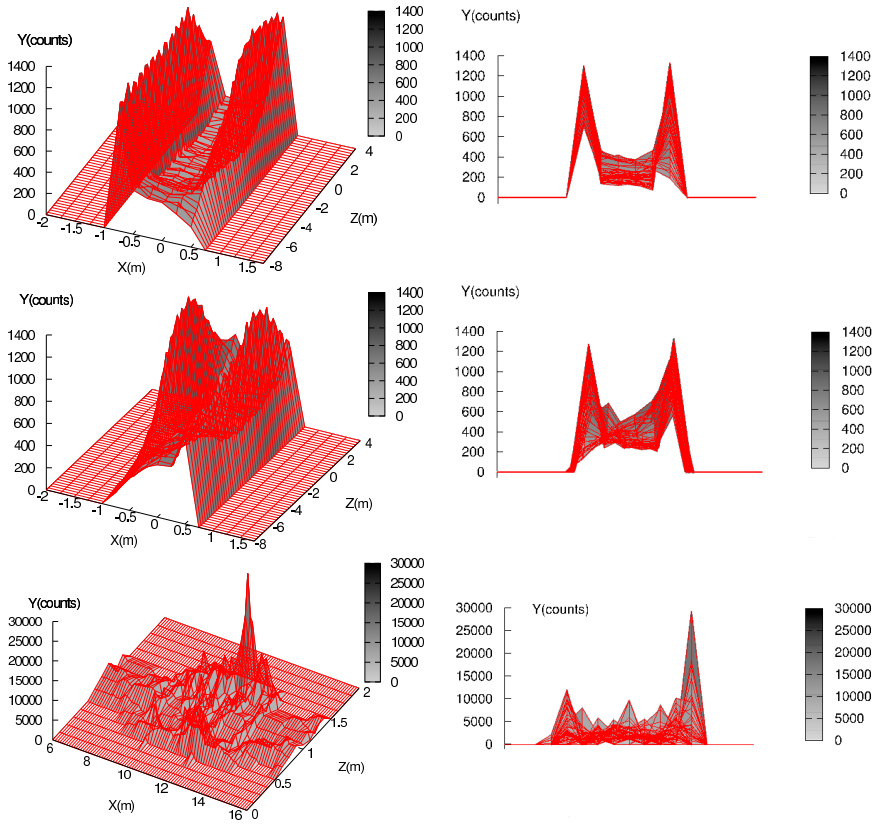


Figure 9: Density maps for the crossing inside a corridor. First row perspective and front views of the map for the experiment with Policy-Reuse. Middle row for the experiment without Policy-Reuse. Bottom row for the experiment with the Helbing's model. The data of the Helbing model have been collected from the implementation described in [46]. The configuration is the same that the other two experiments.

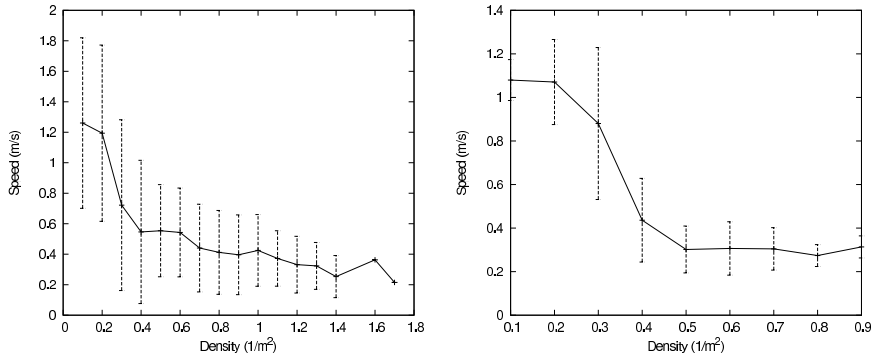


Figure 10: Fundamental diagrams of the crossing experiment for a measure point in the center of the corridor with 8 agents. Left: MARL-Ped experiment. Right: Helbing's experiment.

	Individual average performance percentage	Collective average performance percentage
MARL-Ped with Policy-Reuse	96.0 ± 0.8	80.0 ± 3.1
MARL-Ped without Policy-Reuse	92.9 ± 1.0	67.9 ± 4.2

Table 2: Performance analysis in simulation for the crossing experiments with MARL-Ped. The individual performance percentage counts the times (of 100 episodes) that an agent has reached to the goal. The collective performance measures the number of times (of 100 episodes) in which all the agents arrived at the goal. The figures are averages of 10 experiments.

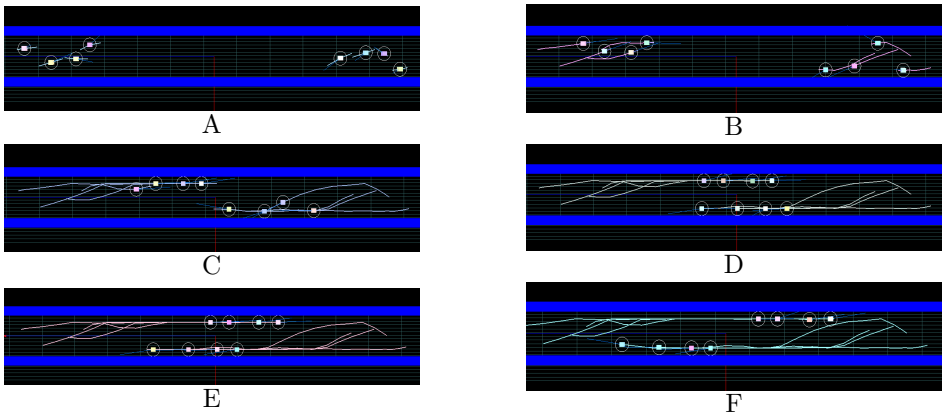


Figure 11: A simulation in the experiment of crossing in a corridor. The temporal sequence of stills is sorted alphabetically.

(those in which all the agents arrive at the corresponding goal). On the contrary, the influence is not important at the individual level, stressing the coordination role of the Policy-Reuse transfer technique in the learning process. In the case of the Helbing experiment all the episodes are successful.

In Figure 11 several images of a simulation are displayed as a sequence. Note the anticipatory organization in lanes in the images B and C (that is, the agents do not wait for the imminence of a crash to organize as occur with models based on forces or potential fields). The videos of several simulations can be seen at the URL http::/www.uv.es/agentes/RL. The reader can find a video of special interest that reproduce two simulations of the crossing problem with the Helbing and the MARL-Ped approaches. In this video, the reader can appreciate the different solutions that both approaches give to the problem. In the case of Helbing, the agents tend to occupy the center of the corridor where the collisions occur. In the case of the MARL-Ped, the agents have learnt to anticipate the crossing and two lanes are built before the crossing occurs. Note that the lane formation occurs at the beginning of each simulation but the initial positions of the agents occupy all the wideness of the corridor.

Individual average performance percentage	Collective average performance percentage
98.1 ± 1.0	96.8 ± 1.5

Table 3: Performance analysis in simulation for the labyrinth with MARL-Ped. The individual performance percentage counts the times (of 100 episodes) that an agent has reached to the goal. The collective performance measures the number of times (of 100 episodes) in which all the agents arrived at the goal. The figures are averages of 10 experiments.

7.4. Pedestrians in a maze

In this experiment the use of fundamental diagrams and density maps are not adequate because only two agents are present in the scenario. The visualization of the learned behaviors show that the agents have learned both, a local control and also a global control capable of planning the path to reach the goal and are able to properly combine them. The Figure 12 shows a sequence of an episode in simulation. Note in the image D of the sequence that the agent coming from the left side executes a maneuver to avoid the collision with the other agent, situated near the wall. This control maneuver corresponds to the operational level. The trajectories of the agents are not symmetrical because each agent learns independently and, therefore, the control of the movement is different for each agent.

Table 3 shows the individual and collective percentages of successful episodes. Note the high percentage of success ($> 95\%$) in both performance measures.

8. Conclusions and future work

In this paper we explore the capabilities of our RL-based framework for pedestrian simulation (MARL-Ped) in three different paradigmatic situations. The main contribution of this work is the empirical demonstration that RL techniques are capable of converging to policies that generate pedestrian behaviors. Our framework solves the navigation problems at different pedestrian behavior levels (strategical, tactical and functional) on the contrary to other pedestrian models. Thus, the shortest path *vs* quickest path scenario shows capabilities to solve navigation problems at the tactical level, because the learned behaviors reproduce the situation of the election between the shortest or the quickest path. The results of the labyrinth experiment demonstrate capabilities of the behaviors at the strategical level combining local and global navigation. In the crossing scenario the emergent behaviors (lanes formation) have been created. Besides, the specific benefits and characteristics of the RL approach described in the introduction and in Section 3, make it an interesting alternative for the simulation of pedestrian groups.

The comparison of the fundamental diagrams and density maps with the Helbing's social forces model, reveals similarities in the generated pedestrian dynamics. As the Helbing's model is a well-known pedestrian model, these similarities reinforce the idea that the agents of our simulator behave like pedestrians and not as another different vehicular embodied agent.

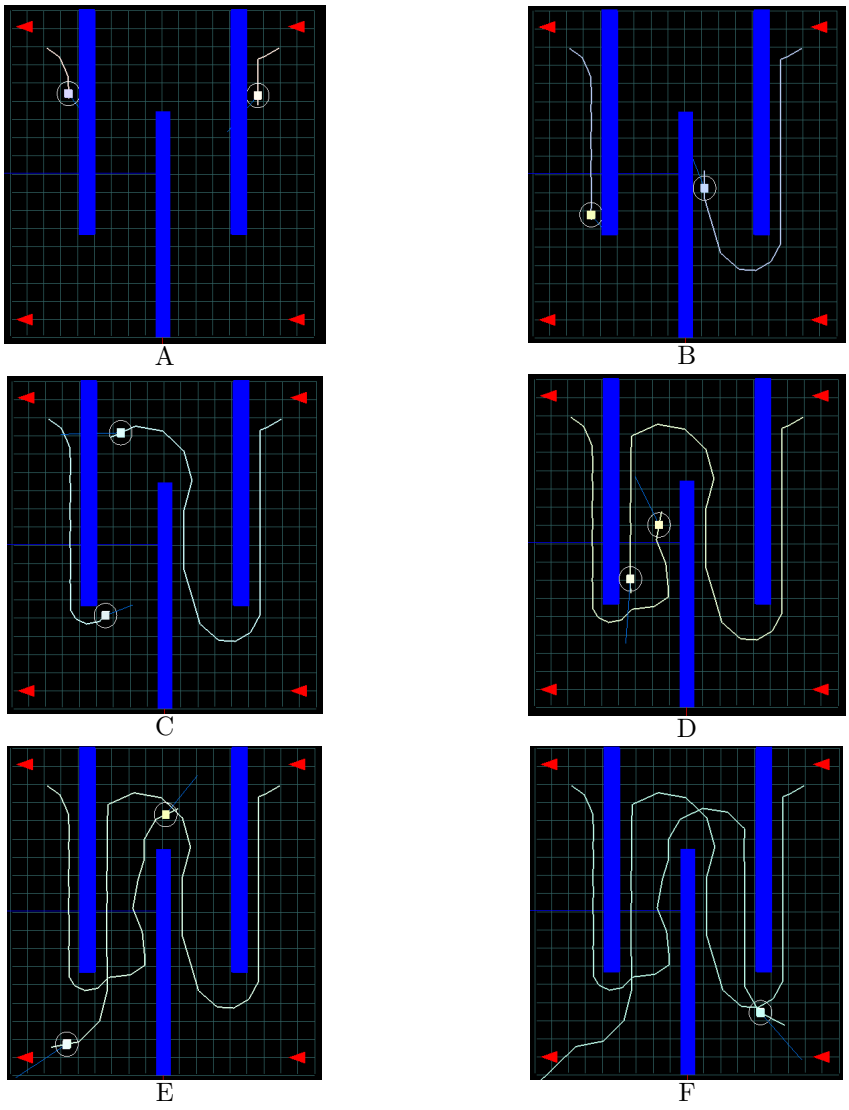


Figure 12: A simulation in the experiment of the labyrinth. The temporal sequence of stills is sorted alphabetically. The goals are situated at the bottom of the maze

A drawback in our approach is the lack of editability or authoring of the learned behaviors. The learning process is carried out autonomously and without supervision by each agent. The result is a learned value function which is not easy of modify, that constitutes an important issue in behavioral animation for virtual environments.

Once proved the usefulness of the RL techniques in the pedestrian simulation domain, other scenarios in which the studied emergent collective behaviors appear will be tested. For instance, a group of agents turning a corner (in the the shortest *vs.* quickest problem), and scenarios with higher densities and obstacles like columns (in the crossing problem).

The problem of authoring (the capability of the user/author to control the final animation) has to be situated during the learning process and not after it has finished. To achieve a more authorable/controllable model, RL techniques like learning from demonstration or reward shaping will be tested in a future work.

Acknowledgements

The authors want to acknowledge to Dr. Illés Farkas, who kindly provided us the code for the Helbing's corridor experiment and helped us with the configuration parameters.

This work has been partially supported by the University of Valencia under project UV-INV-PRECOMP13-115032, the Spanish MICINN and European Commission FEDER funds under grants Consolider-Ingenio CSD2006-00046, TIN2009-14475-C04-04, TRA2009-0080. Fernando Fernández is supported by grant TIN2012-38079-C03-02 of Ministerio de Economía y Competitividad.

References

[1] P. Gipps, B. Marsjo, A microsimulation model for pedestrian flows, Math Comp Sim 27 (1985) 95–105.

[2] C. Reynolds, Evolution of corridor following behavior in a noisy world, in: From animals to animats. Proceedings of the third international conference on simulation of adaptive behavior, MIT Press, 2003, pp. 402–410.

[3] W. Shao, D. Terzopoulos, Autonomous pedestrians, in: Proceedings of the 2005 ACM SIGGRAPH symposium on Computer animation, ACM Press, New York, NY, USA, 2005, pp. 19–28.

[4] D. Helbing, P. Molnár, Social force model for pedestrian dynamics, Physics Review E 51 (1995) 4282–4286.

[5] N. Pelechano, J. Allbeck, N. Badler, Controlling individual agents in high-density crowd simulation, in: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ACM, New York, 2007, pp. 99–108.

[6] W. Daamen, Modelling passenger flows in public transport facilities, Ph.D. thesis, Delft University of Technology, The Netherlands (2004).

[7] T. Robin, G. Antonioni, M. Bierlaire, J. Cruz, Specification, estimation and validation of a pedestrian walking behavior model, Transportation Research 43 (2009) 36–56.

[8] C. Reynolds, Steering behaviors for autonomous characters, in: Game Developers Conference, Miller Freeman Game Group, San Francisco, California., 1999, pp. 763–782.

[9] S. Patil, J. V. D. Berg, S. Curtis, M. Lin, D. Manocha, Directing crowd simulations using navigation fields, IEEE Trans. on Visualization and Computer Graphics 17 (2) (2011) 244–254.

[10] A. Treuille, S. Cooper, Z. Popovic, Continuum crowds, ACM Transactions on Graphics (TOG) 25 (3) (2006) 1160–1168.

[11] M. Sung, M. Gleicher, S. Chenney, Scalable behaviors for crowd simulations, in: SIGGRAPH'04 symposium on Computer animation, ACM Press, 2004, pp. 519–528.

[12] H. Xi, Y.-J. Son, Two-level modeling framework for pedestrian route choice and walking behaviors, Simulation Modelling Practice and Theory 22 (2012) 28–46.

[13] D. Helbing, L. Buzna, A. Johansson, T. Werner, Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions, Transportation Science 39 (1) (2005) 1–24.

[14] T. I. Lakoba, D. J. Kaup, N. M. Finkelstein, Modifications of the helbing-molnár-farkas-vicsek social force model for pedestrian evolution, Simulation 81 (5) (2005) 339–352.

[15] D. Parisi, C. Dorso, Morphological and dynamical aspects of the room evacuation process, Physica A: Statistical Mechanics and its Applications 385 (1) (2007) 343 – 355.

[16] D. O'Sullivan, M. Haklay, Agent-based models and individualism: is the world agent-based?, Environment and Planning A 32 (2000) 1409–1425.

[17] N. Shiwakoti, M. Sarvi, G. Rose, M. Burd, Animal dynamics based approach for modeling pedestrian crowd egress under panic conditions, Transportation Research Part B: Methodological 45 (9) (2011) 1433 – 1449.

[18] J. Pettre, J. Ondrej, A. Olivier, A. Creutal, S. Donikian, Experiment-based modeling, simulation and validation of interactions between virtual walkers, in: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ACM, New York, 2009, pp. 189–198.

[19] S. Hoogendoorn, P. Bovy, Pedestrian route-choice and activity scheduling theory and models, Transportation Research Part B: Methodology 38 (2004) 169–190.

[20] S. Guy, S. Curtis, M. Lin, D. Manocha, Least-effort trajectories lead to emergent crowd behaviors, Physics Review E 85 (2012) 0161100–0161107.

[21] G. K. Zipf, Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology, Addison-Wesley, 1949.

[22] K. Still, Crowd dynamics, Ph.D. thesis, Department of Mathematics. Warwick University, UK (August 2000).

[23] D. Helbing, A. Johansson, Pedestrian, crowd and evacuation dynamics, in: Encyclopedia of Complexity and Systems Science, Vol. 16, Springer, 2010, pp. 6476–6495.

[24] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998.

[25] J. Lee, K. H. Lee, Precomputing avatar behavior from human motion data, in: Eurographics/ACM SIGGRAPH Symposium on Computer Animation. R. Boulic, D.K. Pai (Eds.), 2004, pp. 79–87.

[26] J. McCann, N. S. Pollard, Responsive characters from motion fragments, ACM Transactions on Graphics (SIGGRAPH 2007) 26 (3).

[27] A. Treuille, Y. Lee, Z. Popović, Near-optimal character animation with continuous control, ACM Transactions on Graphics (SIGGRAPH 2007) 26 (3).

[28] L. P. Kaelbling, M. L. Littman, A. W. Moore, Reinforcement learning: A survey, Int. Journal of Artificial Intelligence Research 4 (1996) 237–285.

[29] The MPI Forum, MPI: A message passing interface (1993).

[30] M. Bierlaire, T. Robin, Pedestrians choices, in: H. Timmermans (Ed.), Pedestrian Behavior, Emerald, 2009, pp. 1–26.

[31] F. Martinez-Gil, M. Lozano, F. Fernández, Calibrating a motion model based on reinforcement learning for pedestrian simulation., in: Motion In Games (MIG'12)). LNCS 7660. Springer., 2012, pp. 302–313.

[32] C. Szepesvári, Algorithms for reinforcement learning, Morgan Claypool, 2010.

[33] J. S. Albus, A new approach to manipulator control: the cerebellar model articulation controller (CMAC), Journal of Dynamic Systems, Measurement, and Control 97 (1975) 220–227.

[34] G. A. Rummery, M. Niranjan, On-line q-learning using connectionist systems, Tech. Rep. CUED/F-INFENG/TR 166, Engineering Departement, Cambridge University (1994).

[35] F. Martinez-Gil, M. Lozano, F. Fernández, Multi-agent reinforcement learning for simulating pedestrian navigation, in: Adaptive and Learning Agents - International Workshop, ALA 2011, Held at AAMAS 2011, Taipei, Taiwan, May 2, 2011, Revised Selected Papers, Vol. 7113 of Lecture Notes in Computer Science, Springer, 2012, pp. 54–69.

[36] D. Helbing, P. Molnár, I. Farkas, K. Bolay, Self-organizing pedestrian movement, Environment and Planning. Part B: Planning and Design 28 (2001) 361–383.

[37] A. Johansson, T. Kretz, Applied pedestrian modeling, in: A. Heppenstall, A. Crooks, L. See, M. Batty (Eds.), Spatial Agent-based Models: Principles, Concepts and Applications, Springer, 2011.

[38] T. Kretz, Pedestrian traffic: on the quickest path, Journal of Statistical Mechanics: Theory and Experiment 3 (2009) P03012.

[39] D. Helbing, T. Vicsek, Optimal self-organization, New Journal of Physics 1 (1999) 13.1–13.17.

[40] F. Fernández, M. Veloso, Learning domain structure through probabilistic policy reuse in reinforcement learning, Progress in Artificial Intelligence 2 (1) (2013) 13–27.

[41] U. Weidmann, Transporttechnik der fussgänger - transporttechnische eigenschaften des fussgngerverkehrs (literaturstudie), Literature Research 90, IVT an der ETH Zürich, ETH-Hönggerberg, CH-8093 Zürich (1993).

[42] D. Helbing, A. Johansson, Pedestrian, crowd and evacuation dynamics, in: Encyclopedia of Complexity and Systems Science, Springer, 2009, pp. 6476–6495.

[43] A. Seyfried, B. Steffen, W. Klingsch, M. Boltes, The fundamental diagram of pedestrian movement revisited, Journal of Statistical Mechanics: Theory and Experiment (2005) P10002.

[44] D. Helbing, A. Johansson, H. Z. Al-Abideen, Dynamics of crowd disasters: An empirical study, Phys. Rev. E 75 (2007) 046109.

[45] D. Helbing, I. J. Farkas, T. Vicsek, Simulating dynamical features of scape panic, Nature 407 (2000) 487.

[46] D. Helbing, I. J. Farkas, T. Vicsek, Freezing by heating in a driven mesoscopic system, Physical Review Letters 84 (2000) 1240–1243.