


ACMICS: an agent communication model for interacting crowd simulation

Kurtulus Kullu^{1,3} · Uğur Güdükbay¹  ·
Dinesh Manocha²

Published online: 12 May 2017
© The Author(s) 2017

Abstract Behavioral plausibility is one of the major aims of crowd simulation research. We present a novel approach that simulates communication between the agents and assess its influence on overall crowd behavior. Our formulation uses a communication model that tends to simulate human-like communication capability. The underlying formulation is based on a message structure that corresponds to a simplified version of Foundation for Intelligent Physical Agents Agent Communication Language Message Structure Specification. Our algorithm distinguishes between low- and high-level communication tasks so that ACMICS can be easily extended and employed in new simulation scenarios. We highlight the performance of our communication model on different crowd simulation scenarios. We also extend our approach to model evacuation behavior in unknown environments. Overall, our communication model has a small runtime overhead and can be used for interactive simulation with tens or hundreds of agents.

Keywords Crowd simulation · Communication model · Agent communication · Foundation for Intelligent Physical Agents (FIPA) · Agent Communication Language (ACL)

Electronic supplementary material The online version of this article (doi:[10.1007/s10458-017-9366-8](https://doi.org/10.1007/s10458-017-9366-8)) contains supplementary material, which is available to authorized users.

✉ Uğur Güdükbay
gudukbay@cs.bilkent.edu.tr

Kurtulus Kullu
kullu@cs.bilkent.edu.tr

Dinesh Manocha
dm@cs.unc.edu

¹ Department of Computer Engineering, Bilkent University, Ankara, Turkey

² Department of Computer Science, University of North Carolina, Chapel Hill, NC, USA

³ Department of Computer Engineering, Ankara University, Ankara, Turkey

1 Introduction

Virtually simulating crowds (of people or other agents) is an active area of research in computer graphics, virtual environments, and artificial intelligence (AI). Crowd simulation algorithms are widely used to generate plausible effects in computer animation and games. At the same time, they are used to predict the pedestrian flow in architectural models and urban environments [14].

One of the major goals in the field is to automate the simulation of behaviorally plausible crowds. Crowd heterogeneity and emergent agent behaviors are two important components that are often targeted to achieve this goal [13, 34]. These problems have been extensively studied and techniques have been proposed that make use of cognitive, behavioral, and psychological models [9, 11, 13, 19, 51].

Our work is motivated by the fact that communication in general is inherent in real-world crowds. Consider any crowd that is formed naturally, such as concert or sports spectators or people evacuating a building. In these scenarios, the behavior and movement of the crowd can be influenced by the information shared between the individuals. However, communication between the agents has not received much attention in the crowd simulation literature. In this paper, we mainly address the problem of modeling the effects of deliberate inter-agent communication as part of interactive crowd simulation.

Limiting ourselves to deliberate communication is a key issue in this problem. If a broader communication definition such as “transfer of information” is used, then almost everything can be included as part of “communication”. For example, an agent’s entire perception can be thought of as information being transferred from the environment to the agent [50] or internal cognitive processes can be thought of as communicating components (intra-communication). Our communication model focuses on deliberate inter-agent communication. Following other agents, which could be regarded as a form of non-intentional communication, is taken into account as part of the navigation logic.

Main results We present ACMICS, a novel approach to simulate communication between agents in a crowd simulation system, and we evaluate its impact on simulated crowd behaviors. ACMICS makes use of a simplified version of a message structure specification from the multi-agent systems community known as Foundation for Intelligent Physical Agents (FIPA) Agent Communication Language (ACL) Message Structure Specification [35]. FIPA is a standards organization operating under IEEE, which aims to produce software standards specifications for agent-based and multi-agent systems. ACMICS is capable of handling message sending/receiving between the agents of a crowd in a human-like manner. Our approach makes no assumptions about local or global navigation schemes and can be easily combined with them. Some of the novel components of our work include:

1. A novel approach to facilitate inter-agent communication in a crowd simulation system that
 - (a) is designed as a separate module in the agent architecture,
 - (b) requires some form of perception capability,
 - (c) separates low- and high-level tasks in a modular manner, and
 - (d) can be easily extended and used in arbitrary scenarios and/or can support different forms of communication.
2. A high-level planning algorithm to simulate the evacuation behavior in new or unknown environments where the agents do not have a priori knowledge about their environment.

We demonstrate that, based on ACMICS, the agents autonomously communicate to navigate more effectively in such scenarios.

We highlight the application of ACMICS in different scenarios to facilitate deliberate inter-agent communication. Initially, we use it to enable hollow communications, i.e. communications that do not involve important information transfer. Pedestrian flow is measured with and without communication and compared with the results from other simulators. Next, ACMICS is combined with the high-level evacuation navigation logic to highlight its application in facilitating meaningful agent interactions. The effects on pedestrian evacuation times and trajectories are analyzed. Lastly, simulated agent trajectories with/without communication are compared with those extracted from a real crowd video using the vfractal metric [26]. Our approach can be combined with any multi-agent crowd simulation algorithm and has a small runtime overhead. In practice, we can simulate crowds with tens or hundreds of agents at interactive rates on current desktop systems.

The rest of the paper is organized as follows. Section 2 provides a summary of related literature on crowd simulation, the vfractal metric, communication models, and communication between virtual agents. In Sect. 3, we introduce the notation and describe ACMICS, our agent communication approach. We describe various scenarios in Sect. 4 that are used to evaluate our model and highlight the performance.

2 Related work

Ali et al. [1] and Thalmann and Musse [47] provide an overview of crowd simulation algorithms. In [34], a survey of common methods, existing crowd simulation algorithms and systems is provided. At a broad level, these methods can be classified into two categories: *macroscopic* and *microscopic*. Macroscopic methods focus on the crowd as a whole rather than individuals in it. On the other hand, microscopic models concentrate on the behaviors and decisions of individuals as well as their interaction with each other. Further categorizations into groups such as fluid-dynamic or gas-kinetic models [16], social force models [15], cellular automata models [4], velocity-based methods [2], and biomechanic models [12] have also been proposed.

State-of-the-art techniques that populate virtual environments with multiple human-like agents or crowds are used to generate special effects in movies and animation. In the last decade, the challenge has become the real-time generation or simulation of autonomous crowds [46]. These are necessary for games and virtual or augmented reality systems. Autonomy is needed for agents to react to events in real-time.

Some of the approaches for crowd simulation [23,24] focus on the visual plausibility of the simulation while others [10,13,32,33] focus on behavioral plausibility. Many researchers have borrowed concepts from psychology or cognitive literature and applied them to crowd simulation. For example, Durupinar et al. [9] incorporate three elements, namely, *personality*, *emotion*, and *mood*, to an agent model. Kim et al. [19] present an efficient algorithm to model dynamic behaviors in crowd simulation using General Adaptation Syndrome. Silverman et al. [41,42] use different performance moderator functions from behavioral psychology literature to improve the realism of virtual agents.

Menge [8] and ADAPT [40] are two recent extensible modular frameworks aimed at simulating virtual agents. Menge primarily focuses on crowd movement. The crowd simulation problem is decomposed into four subproblems, each of which is to be solved for every agent in the crowd: *goal determination*, *planning*, *facilitating reactive behavior*, and *agent*

motion. The ADAPT framework includes capabilities for character animation, navigation, and behavior. Its primary focus is on animation, in particular, on seamless integration of multiple character animation controllers. It couples a system for blending arbitrary animations with static and dynamic navigation capabilities for human characters. It is possible to combine our communication model with these approaches.

The term *vfractal* [26] is used to refer to a group of methods to estimate fractal dimension [22] for animal movement trajectories. The estimation is an indication of trajectories' straightness/crookedness. The values theoretically range between one for a straight trajectory and two for a trajectory so tortuous that covers a plane. Vfractal estimators are commonly used in biology-related literature for animal movement paths. They have also been used to evaluate agent-based simulation methods [48] and pedestrian egress behavior [27]. The estimations involve dividing a trajectory into pairs of fixed size steps. The same measurements for randomly selected steps are averaged to acquire estimation results at that step size. Estimations are carried out in the same manner for different step sizes. An advantage of vfractal estimators is that confidence values in the estimations can also be calculated.

2.1 Communication

Communication is often studied from the perspectives of other disciplines. Craig [7] argued that an identifiable field of communication theory did not exist despite extensive literature and investigation. Nevertheless, existing theories provide us various models of communication. One of the earliest models of communication was developed by Shannon and Weaver [39] in order to mirror the functioning of radio and telephone technologies. Others, such as Berlo [3], extended their initial model. There are often eight common components considered in such models, which are generally referred to as *transmission models*. These are *source*, *message*, *transmitter*, *signal*, *channel (carrier)*, *noise*, *receiver*, and *destination*. Transmission models are simple, general, and quantifiable. On the other hand, these transmission models are considered as inadequate in terms of modeling real-world human communications [6].

Schramm's [37] works, which are mainly on mass communication and its effects, provide a different view of communication and particularly indicate that the impact of messages (desired or undesired) on the target should be examined. There are two important concepts in his model: *feedback* and *field of experience (FoE)*. FoE is a representation of a communicator's beliefs, experiences, and so on. An individual can only encode and decode messages with respect to her/his own FoE. Individual FoEs of communicators must have an intersection for successful communication. In our approach, we name the higher-level communication layer after this concept of FoE, making a connection with Schramm's model.

2.2 Communication and virtual agents

A research area that brings communication and virtual agents together is Embodied Conversational Agents (ECAs) [5]. The primary aim in this field is to develop autonomous agents that can converse with real people. Some efforts take the content and form of conversation (e.g. what the agent is saying) as their central issues, whereas others' main concerns are facial expressions and gestures. The works in this category are not directly related to our work because they focus on agents communicating with real people through specific forms such as speech, whereas our focus is on inter-agent communication through an abstract form of communication.

Table 1 Parameters in FIPA ACL message structure specification [35]

| | Parameter | Description |
|--|------------------------|--|
| The message structure in our approach uses the parameter subset indicated by the parameter names in italic. The message structure defines what is actually exchanged between the agents and it can be extended as needed | <i>Performative</i> | Type of the communicative act |
| | <i>Sender</i> | Agent sending the message |
| | <i>Receiver</i> | Agent to receive the message |
| | <i>Reply-to</i> | Agent that the replies should be sent to |
| | <i>content</i> | Content of the message |
| | <i>Language</i> | Language of the content |
| | <i>Encoding</i> | Encoding of the content expression |
| | <i>Ontology</i> | Ontology(s) required to understand content |
| | <i>Protocol</i> | Interaction protocol sender is employing |
| | <i>Conversation-id</i> | Conversation identifier |
| | <i>Reply-with</i> | Identifier that should be used in replies |
| | <i>In-reply-to</i> | The message this one responds to |
| | <i>Reply-by</i> | Time by which replies should be received |

Sun et al. [45] present a framework to distribute dialogs among agents in a virtual crowd. In their simulations, the environmental context and agent attributes initiate, guide, and affect the evolution of (unscripted) conversations in a virtual crowd. An important difference between this and our work is that the dialogs in their work are only for visual improvement and they do not include information sharing that could affect behavior.

In another work by Park et al. [31], small groups are formed in a larger crowd in order to improve the realism. These small groups maintain their cohesiveness through communication between its members. Through user studies, they showed that the believability of the animation is improved by communicative and social interactions among virtual characters in their common ground theory-based simulation.

FIPA aims to produce software standards specifications for agent-based and multi-agent systems. Although related, the usage of the term “agent” in these areas and in crowd simulation can be slightly different (cf. [29] for a discussion on this). ACL [35] specification is one of the most widely adopted standards of FIPA. It is a standard language for software agent communications based on speech act theory [38]. FIPA ACL Message Structure Specification standardizes the message form. Table 1 shows the list of message parameters in the specification that can be extended by specific implementations according to the requirements of the application. The parameters that are included in a message are application dependent. The only mandatory parameter is *performative* but most messages are expected to also contain *sender*, *receiver*, and *content* parameters.

A preliminary version of our communication model was presented at CASA’14 as a short paper [20]. This short paper contained a different layered design. The aim at the time was for the communication model to be as human-like as possible and to couple it with a more complex agent design. Aiming for a human-like communication capability (e.g. psychology influencing communication) introduces complexity, but the subtle benefits are not significantly observable in a crowd. As a result, our aim has evolved into developing a less complex and generic communication model (i.e. a model that is easy to combine with other agent architectures and use in new scenarios). Moreover, current work employs the model in a more realistic environment, presents improved (3D) visualization, and provides a more formal evaluation.

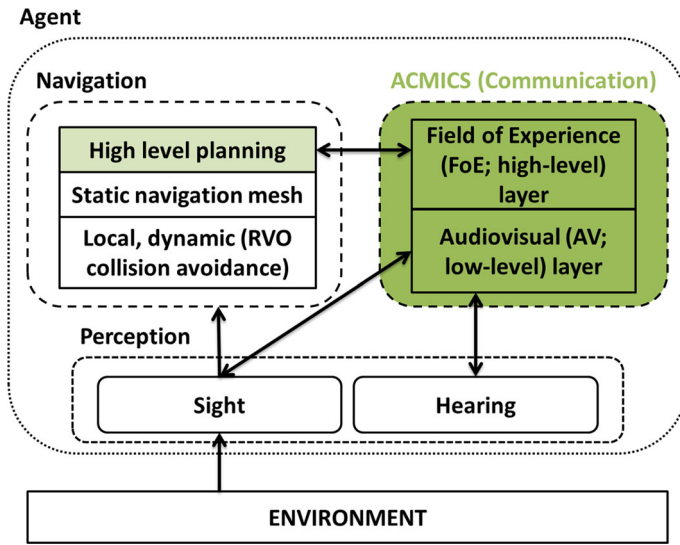


Fig. 1 The agent architecture with three major components (communication, navigation, and perception), their internal structure, and their relationships. The perception subcomponents, sight and hearing, track which other agents are in sight and hearing range. The sight subcomponent also tracks important objects in the environment such as doors. These data are provided to the other components when requested. Navigation includes a local collision avoidance solution (Reciprocal Velocity Obstacles; RVO) and a global path planning solution (navigation mesh) as well as a higher-level, scenario-dependent planning part. Planning can be as simple as deciding a single target position at the beginning or it can be a complex algorithm to simulate decision-making during evacuation (cf. Fig. 7). Navigation does not require the communication component but cooperates with it when it is enabled. Communication separates message/scenario-dependent (high-level, FoE) tasks from those that are independent (low-level, AV, cf. Figs. 3–4). The green color indicates the novel components of our approach (Color figure online)

3 Agent AI and communication model

We model an agent with three major components: *perception*, *communication* (ACMICS), and *navigation* (Fig. 1). The communication component of the architecture is our main contribution. The rest is mainly what is needed in order to apply ACMICS to example scenarios. Navigation is necessary for the agents to move around and perception is needed by the other two components.

Perception consists of *hearing* and *sight* subcomponents. Hearing is modeled with a spherical volume around the agent that represents the hearing range and sight is represented using a pyramid shape in front of the agent (cf. Fig. 2). These subcomponents continuously keep track of important objects such as other agents, doors, or signs that are in hearing and/or sight range. Their primary function is to provide this data to the navigation and communication components on request. This relationship is shown in Fig. 1 by the arrows from the perception components to the other components.

Navigation makes use of well-known existing methods: (i) a precomputed navigation mesh [43] to calculate static obstacle avoiding paths and (ii) Reciprocal Velocity Obstacles (RVO) [2] for avoiding dynamic obstacles (i.e. other agents). This two-layer navigation capability is controlled by a third and higher planning layer. The third layer is scenario dependent. In some examples, its control is simple and straightforward and only involves specifying the final destination. In other examples, however, it can be more complex. For

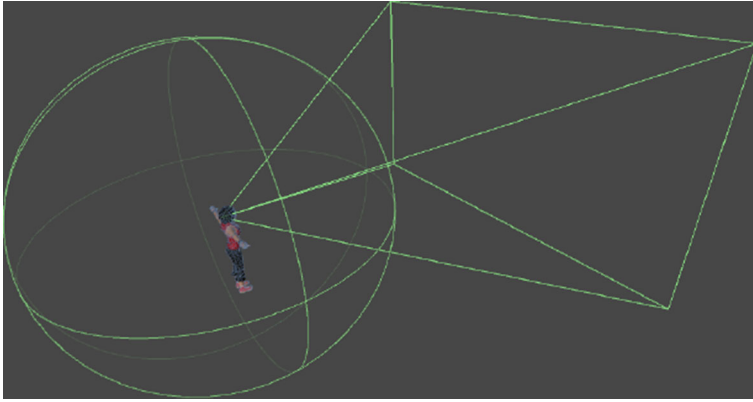


Fig. 2 Visualization of hearing and sight volumes

instance, when we want to simulate an agent’s lack of knowledge, instead of calculating the path between the target and current position, intermediate targets can be used to simulate exploration.

3.1 Communication model

Our focus is on deliberate (i.e. intentional) communication. The reason for this focus is that it is easy to be overinclusive when considering whether or not an interaction is communication. The Oxford dictionary [44] defines communication as:

“the imparting or exchanging of information by speaking, writing, or using some other medium.”

On the surface this definition is clear, but these concepts can become confusing very quickly. Consider the following example. Persons A and B are sitting in the same room and A sees B getting up and turning towards the door to exit. At some point, A has the new information that B is leaving the room but this information was not explicitly communicated from B to A. We can say that this is not communication. However, it could actually be the case that B intends to send a message to A by his/her actions. This example serves to show that real-world communication is a complex phenomenon and it can become difficult to draw a line around what it includes. Also, terms like direct/indirect or implicit/explicit communication are easily used to mean different things. It is not our intention (nor have we the expertise) to argue about what communication is, but for our purposes we draw a line by focusing on deliberate (i.e. intentional) communication. Terms like communication and information are not used to refer to a specific form such as speech. They are considered at an abstract level (communicative intents and meanings in our heads).

We simplify the communication process by making a distinction between low-level and high-level tasks. The low-level tasks, which we call audiovisual (AV) tasks, are independent of the message type. Examples of AV tasks are turning towards the recipient, moving closer if the agent is not close enough, signaling from a distance to catch attention, and so on. In real-world scenarios, actual creation and reception of signals (e.g. sound) are regarded as low-level tasks. However, we are not simulating the signal creation and reception phenomena; these are simply modeled as two AV tasks (*transmit* and *receive*) among others. High-level tasks

actually depend on what the message is and we refer to them as FoE tasks, making a connection with the FoE concept in Schramm's communication model [37]. An example FoE task is responding to a direction request. The separation of AV and FoE layers abstracts away the high-level communication intentions from the low-level modules facilitating communication. As a result of this abstraction, our communication model becomes more generic in the sense that it can be coupled with other agent architectures and used in new scenarios easily. Introducing new types of messages will require additions to FoE layer. Similarly, if a different perception implementation is to be used, the required modifications will be mostly limited to AV tasks.

Another distinction we sometimes make is between *meaningful* and *hollow* communications. We call information exchange that can influence behavior as meaningful communication. For example, in an evacuation scenario, exchanging information about exit routes is an example meaningful communication. On the other hand, we use the term hollow communication to indicate communication in which there is no actual information exchange to influence behavior. For instance, in a scenario such as pedestrians in a street, we may want agents to talk to each other but this talk may not contain any information to affect behavior. The separation of FoE and AV layers allows us to use the same low-level routines for both meaningful and hollow communications. The difference between them will be local to the FoE layer.

The message structure specifies the form of exchanged information. FIPA ACL (cf. Table 1) was adopted partially as needed. Our message types correspond to options for the performative parameter. In addition to *type* (i.e. performative), we use *source*, *destination*, and *content* parameters. We created message types as needed in our example scenarios. The current, extensible list of message types used in our formulation includes:

- WAVE: used to catch recipient's attention when the sender is in sight but not in hearing range;
- CHAT: a message type without any important information content (to simulate hollow communication);
- DIRECTION_REQ: used to ask about the direction, optionally, by providing a target location;
- PATH: used to reply to a DIRECTION_REQ by providing a full path if it can be calculated;
- FINAL_AND_NEAR_TARGET: another reply to a DIRECTION_REQ providing a final target and a nearby intermediate one;
- EXIT_THROUGH: directs to a specific door or staircase;
- FAILURE: the agent is unable to answer with useful information.

This list can be extended as necessary when the communication model is to be applied in new scenarios.

The low-level message sending routine in the AV layer can be summarized with a flowchart (Fig. 3). In this flowchart, *Start* represents the situation in which the agent has a message to be sent to a particular receiver. The first decision requires information from the sender's own perception component. If the receiver is not in sight (i.e. there is no line-of-sight), exploration to find the receiver is required. When the receiver is in sight, new decisions require information from the receiver's perception component. If the sender is within hearing range of the receiver, then the message can be sent directly. Otherwise, the sender has to move before sending the message. In this case, there are two possible situations: If the sender is in sight of the receiver but not in hearing range, the receiver's attention is caught (with a WAVE message) and the sender moves towards the receiver until the receiver is in hearing range. On the other hand,

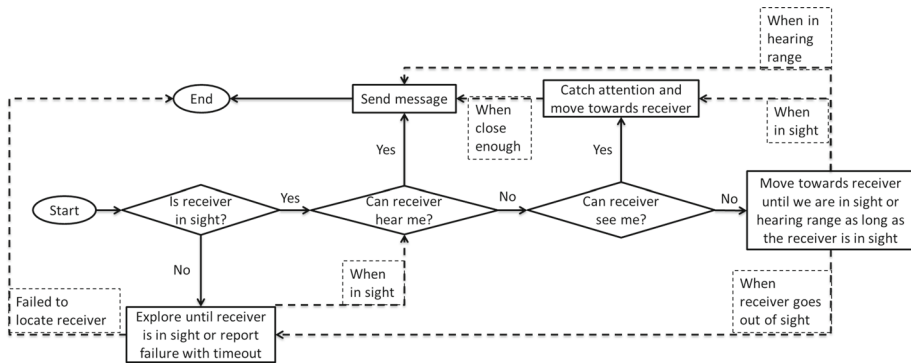


Fig. 3 Message sending routine in the AV layer of ACMICS. Being at *Start* means there is a message to be sent by this agent. Message/scenario type is irrelevant. The agent follows this routine to one of two possible outcomes: sending the message or failure to do so

if the sender is completely outside the receiver’s perception, the sender moves towards the receiver until one of the other cases is met.

ACMICS is implemented as a push-down automaton (PDA) [17]. PDA is chosen for two main reasons. First of all, representing a communicative situation for an agent in terms of communication states is natural and easy. Secondly, backtracking is required in certain situations (e.g. agent needs to roll back to sending the original message after waving to catch the receiver’s attention). The stack in the PDA allows us to keep track of state changes so that rolling back is possible when necessary. The states of the PDA for the AV layer are:

- NOT_COMMUNICATING,
- WANTS_TO_COMMUNICATE,
- FOUND_COMM_TARGET,
- IN_HEARING_RANGE_OF_TARGET,
- WAITING_FOR_TRANSMISSION,
- WAITING_FOR_TRANSMISSION_MOVING_CLOSER,
- IN_SIGHT_OF_TARGET,
- MOVING_CLOSER,
- NOT_IN_SIGHT_OR_HEARING_RANGE,
- RECEIVED_A_MESSAGE, and
- ATTENTION_CAUGHT.

The state names are mostly self-explanatory. The two *WAITING_FOR_TRANSMISSION** states represent the situations where the receiver is busy. Fig. 4 shows the state diagram for the AV layer. *Any State* is a placeholder to represent all other states. A transition from *Any State* to another state *S* means that there is a transition from all states to state *S*. For example, the transition from *Any State* to *RECEIVED_A_MESSAGE* enables the agent to receive a message while in any other state. The transitions are often conditions that would be naturally expected. For instance, the transition from *WANTS_TO_COMMUNICATE* state to *FOUND_COMM_TARGET* state occurs when the agent sees the target agent with whom it wants to communicate.

The state machine is extended with the FoE layer tasks according to the needs of a scenario. We make use of two FoE layer states, namely *DIRECTION_REQUESTED* and *IN_CHAT*, which are specific to the message type and scenario.

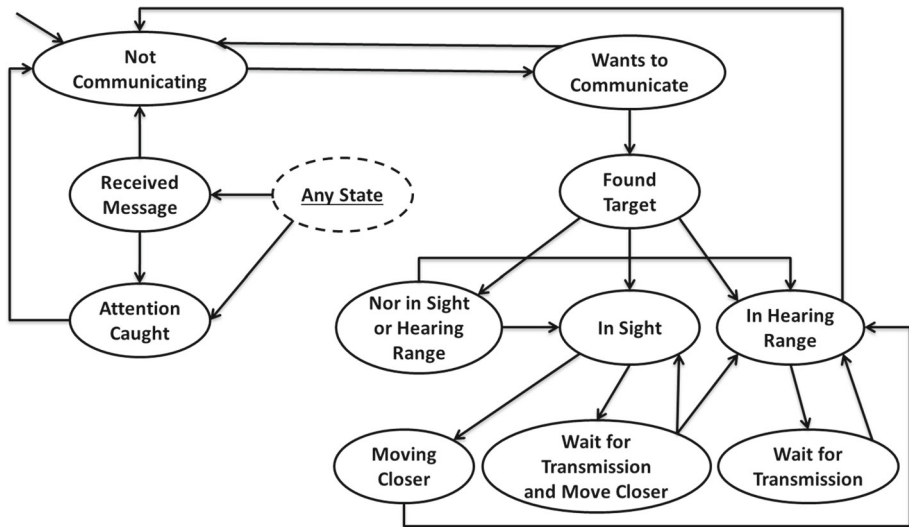


Fig. 4 The state diagram for the AV layer part of ACMICS. Transition labels are omitted for clarity. Any State is a placeholder to represent all other states. The states related to message reception are on the *left side* and those related to message sending (cf. Fig. 3) are on the *right*

4 Multi-agent simulation and validation

We highlight the application of ACMICS by using four example scenarios (cf. Fig. 5) and present our results. The first two scenarios are simpler in the sense that there is only hollow communication. The third scenario is an evacuation scenario using a realistic environment. In the third scenario, ACMICS is employed to simulate meaningful communications such as asking and answering about exit routes. The last scenario is used to compare simulated trajectories with pedestrian trajectories extracted from a video. All simulations were implemented and generated using the Unity[®] Game Engine [49].

The two lower parts in the navigation component in Fig. 1 are scenario independent. In other words, once high-level planning produces a destination to reach, it doesn't make a difference whether this destination is an intermediate target or a final one. Any movement between two positions is achieved by querying the global static navigation mesh and following the returned path with local collision avoidance. These components are implemented using the built-in tools of the game engine. The only agent attribute related to these parts is a preferred speed value for path following and local collision avoidance.

ACMICS, the communication component in our agent model, contains both scenario-dependent and scenario-independent parts. The AV layer tasks are general and low-level routines, but the FoE layer uses them in a scenario-dependent fashion. In fact, specifying the scenario-dependent parts is the main task required to use ACMICS in a new scenario. In a new scenario, a user of ACMICS needs to determine the new message types and the high-level behavior associated with each message type.

4.1 Bidirectional flow and passageway scenarios

The high-level planning part of the navigation component is one of the parts that includes scenario-specific implementations. In the first two scenarios, this part is straightforward in

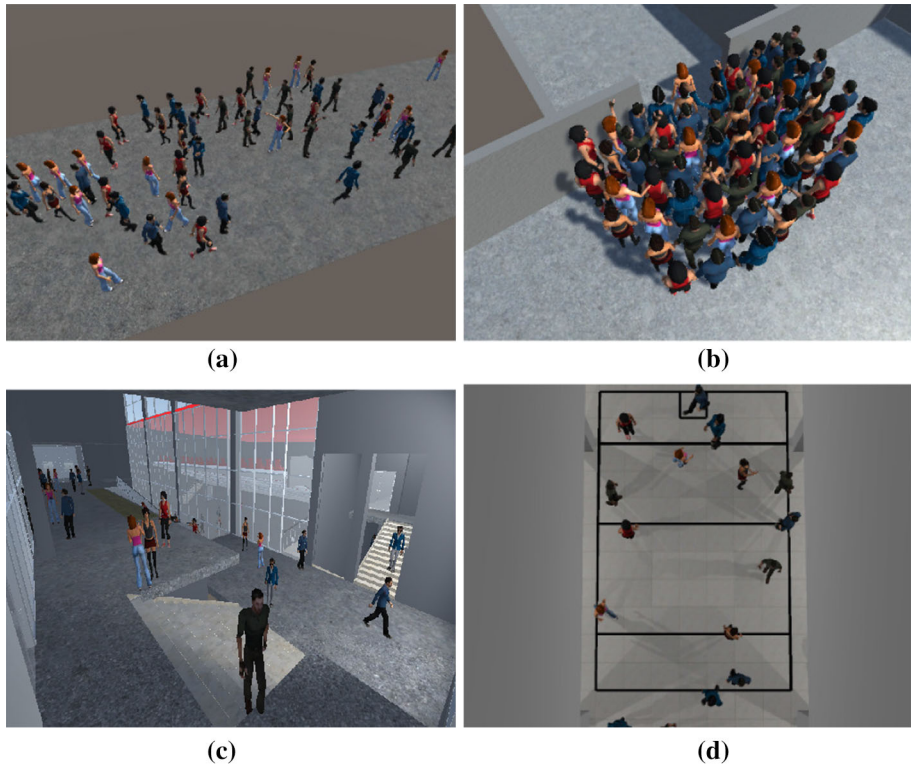


Fig. 5 Screenshots from simulations corresponding to the four example scenarios. Bidirectional flow, passageway, and chat scenarios include only hollow communications (i.e. communications without important information sharing). The evacuation scenario includes meaningful communications where agents share knowledge about the environment. **a** Bidirectional flow. **b** Passageway. **c** Evacuation. **d** Chat

the sense that planning involves only calculating a single destination. In the bidirectional flow scenario, (cf. Fig. 5a) an agent at one end of the street calculates a destination at the other end, and in passageway one (cf. Fig. 5b), all agents calculate a destination at the other side of the passageway down a short corridor.

In both of these scenarios, there is only hollow communication. Hence, the only FoE layer message type is CHAT. Using ACMICS in these scenarios requires determining when and how an agent will send a CHAT message and what to do when it is received. Two parameters, *communication probability* (p_c) and *walk and talk ratio* (p_r), control when and how to send a CHAT message. The parameter p_c is the probability that an agent will engage in communication when the following four conditions are met.

- i) the agent is not already communicating,
- ii) there is another agent within the hearing range,
- iii) this agent has not communicated with the other agent previously, and
- iv) the other agent is not already communicating.

If these conditions are met and the agent decides to engage in communication with probability p_c , another probabilistic decision is made using the second parameter, p_r . The parameter p_r represents the likelihood that the agent will prefer walking and talking to stopping and talking. If the agent decides to do so and the other agent is walking in approximately the

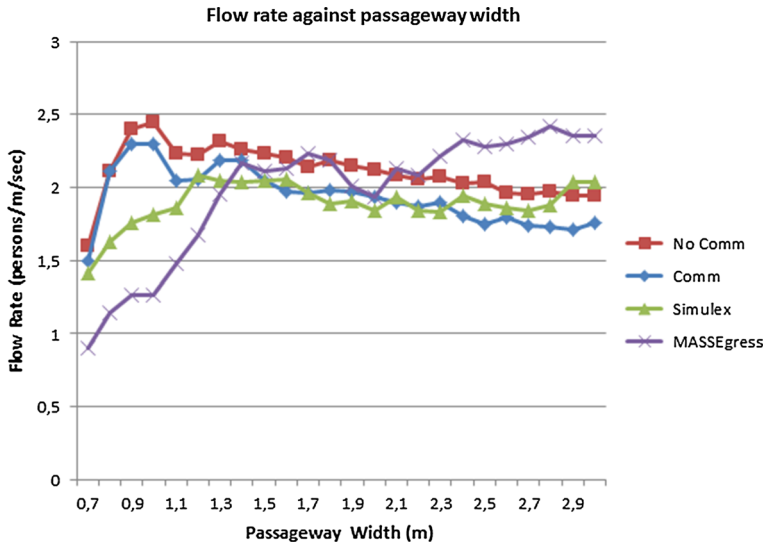


Fig. 6 Comparison of flow rates in the passageway scenario. Note that the flow is calculated as the number of persons per second and per unit width. Because it is ‘per unit width,’ it does not increase as the passageway width increases but rather converges to a value that depends on the preferred speeds. *No Comm* and *Comm* represent our results, whereas *Simulex* and *MASSEgress* represent the results from [30]

same direction, a CHAT message is formed, the preferred speed is adjusted according to the receiver’s speed, and the control is passed down to the AV layer with an indicator for the walk-and-talk type of communication. In the event that the second decision using p_r is negative, the message is formed and the control is passed down without the walk-and-talk indicator. The result is that agents engage in communication by stopping and turning towards each other.

The passageway scenario is used to collect flow data, which is compared to the similar existing data [30]. In his dissertation, Pan takes a psychological and sociological view of human behavior in emergencies to develop a system called MASSEgress. For validation, he repeats a passageway scenario from a previous work that uses Simulex [18]. There are 100 agents in a $5m \times 5m$ area and a single passageway in this scenario. He compares the flow rates for different passageway widths with results from Simulex. We calculate the flow rates in the same way. We ran the simulation twice, first the communication is disabled, and then it is enabled. A summary of the results is given in Fig. 6.

Since the flow rate is calculated as the number of people per second and per unit width, when the passageway becomes larger and allows easier movement, this rate converges to a value that depends on the average walking speed. This pattern is observed in the results for all simulations (the right side of the chart in Fig. 6). When the passageway width is small (the left side of the chart), the results differ more. A narrow passageway causes congestion to worsen and the effect of having different collision avoidance methods becomes amplified. Nevertheless, our results are overall similar to MASSEgress and Simulex results and it is important to note that enabling communication does not affect the flow rate significantly.

4.2 Evacuation scenario

The evacuation scenario (cf. Fig. 5c) differs from the other two scenarios in a few ways. First, the high-level planning in the evacuation scenario is more complex. It includes an algorithm

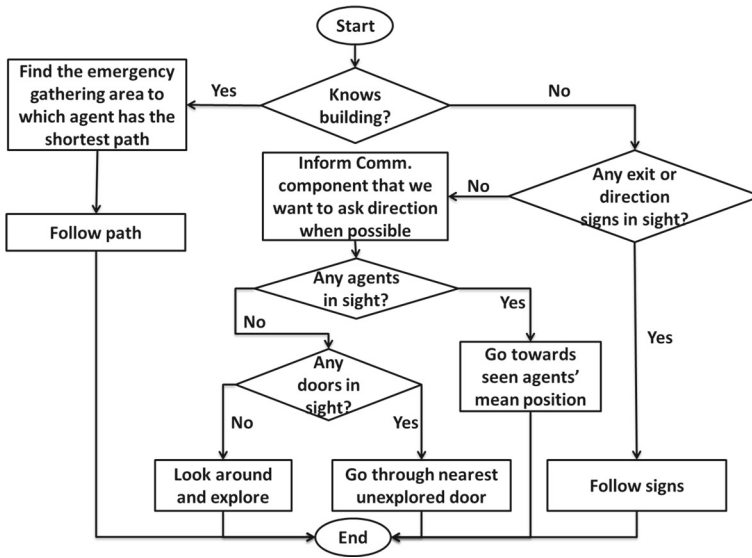


Fig. 7 Our high-level planning algorithm used in the evacuation scenario. The knowledge of the environment is a boolean agent attribute. Direction signs' effects can be turned on/off. This process does not require the communication component. On its own, it can produce plausible evacuation behavior for agents without a priori knowledge about the environment. However, it can also cooperate with the communication component to simulate meaningful communication

we developed to model the evacuation behavior of agents without a priori knowledge about the environment. The overall navigation routine in this scenario is summarized in Fig. 7. A boolean agent attribute represents whether or not an agent knows the environment. If the agent knows the environment, it chooses the emergency gathering area to which it has the shortest path as its destination. In this case, the agent can be communicated with but it does not ask others about how to exit. On the other hand, when the agent does not know about the environment, targeting an exit in sight or following direction signs are the first set of appropriate behaviors for an agent. It is possible to turn off the effect of direction signs for testing purposes. When there are no active signs in sight, the agent wants to communicate with another agent, i.e. ask for direction, when possible. However, navigation continues to run until communication is possible. At this point, the next possible behavior is following others. In the most extreme case that the agent cannot do any of these actions, it displays exploratory behavior.

This process is in control while the agent is in NOT_COMMUNICATING or WANTS_TO_COMMUNICATE states. When the agent is being communicated with, or when it finds a target for communication, the navigation is paused and the communication component takes control. Eventually, a communication instance may cause a change in the agent's knowledge about a path or other agents and the navigation is resumed under these new conditions.

A realistic and complex 3D school building model is used as the 3D model to be evacuated (cf. Fig. 8). A given number of agents can be randomly generated on the navigable areas. We ran simulations with six different settings listed below:

- *Only Nav (Know = 0)*: In this setting, communication and direction signs are disabled. Agents rely only on their navigation capabilities. In addition, none of the agents know the building. In a way, this setting represents the worst case.



Fig. 8 The building model used in evacuation simulations

- *Only Nav (Know = 0.5)*: Similar to the previous one. The difference is that an agent has 0.5 probability of knowing the building; i.e. stochastically, about 50% of agents know the building.
- *Comm*: Communication is enabled but direction signs are still disabled. The building knowledge probability is again 0.5.
- *Sign*: Communication is disabled but use of direction signs is enabled. The building knowledge probability is again 0.5.
- *Comm&Sign*: Both communication and direction sign usage are enabled. The building knowledge probability is again 0.5.
- *Only Nav (Know = 1)*: Similar to the first two settings, but this time agents know the building with probability 1. In other words, all agents can directly calculate their paths. Therefore, this represents the best case in terms of fast evacuation.

Figure 9 gives a summary of measurements for agents' evacuation times. In these charts, the blue marks show the evacuation time averaged over all agents after running the simulation five times. The orange ones are the average of the largest (i.e. worst) five evacuation times. The leftmost and rightmost values in both graphs show what is meant by the worst and best cases, respectively. Remember that when there is no transfer of information and none of the agents know the layout of the building (*Only Nav (Know = 0)*), they rely on seeing an exit, following others, and exploration. Hence, the average evacuation times (blue) are relatively high, i.e. agents take longer to evacuate, in both first columns as expected. So are the averages of the five largest time values (orange). On the other hand, the rightmost values are measured when everyone knows the ideal path to exit the building (*Only Nav (Know = 1)*). As a result, it is clearly observed that both averages are relatively small in these cases.

In the remaining cases, a subset of agents knows the building and, naturally, the values are mostly in between the two cases on each side. The differences resulting from different settings are overall consistent with our expectations. First of all, in general, the more agents there are, the larger the evacuation times become. This increase is normal because of the increased congestion. Also, we see relatively smaller values in *Sign* cases. The use of direction signs naturally reduces the evacuation times. A third notable difference is that the inter-agent communication (*Comm*) seems to cause an increase in evacuation times, especially the large (orange) ones. This increase is also expected and can be explained by the fact that inter-agent communication requires agents to occasionally stop and talk.

If stopping and talking is truly the reason for longer evacuation times, travel distances with inter-agent communication should not be greater than without it. To prove this, we measure the ten longest trajectories in each case. Figure 10 shows the averages for these measurements. It is clear that values in both *Comm* and *Comm&Sign* cases are smaller

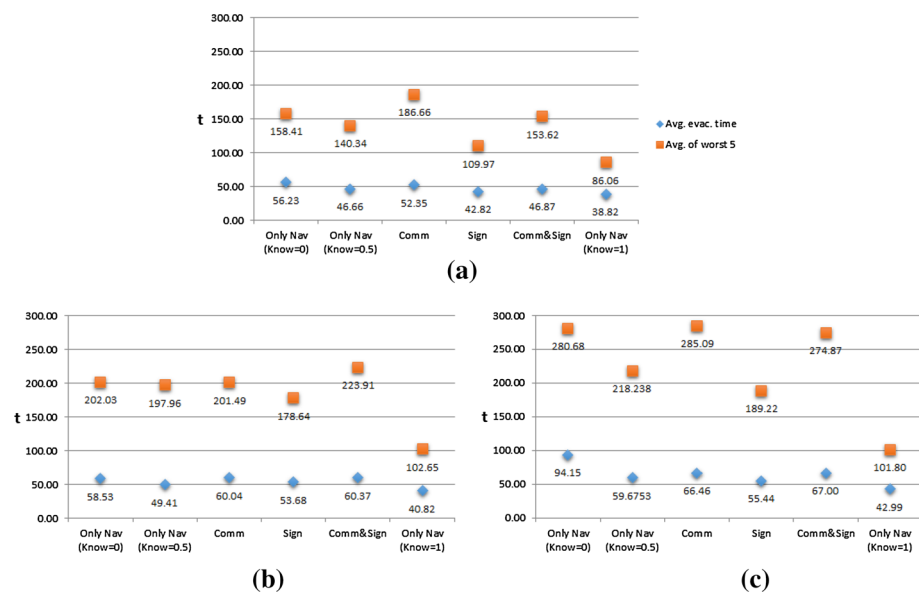


Fig. 9 Average evacuation times measured in the evacuation scenario for 50, 100, and 200 agents placed randomly. There are six different simulation settings: Only Nav (Know=0), Only Nav (Know=0.5), Comm, Sign, Comm&Sign, and Only Nav (Know=1). The simulation is carried out five times for each setting. *Avg. evac. time* (blue) is the average of all agents' evacuation times over all five executions. *Avg. of worst 5* (orange) is the average of the five largest (i.e. worst) evacuation times. The setting names are explained in the text in more detail. Briefly, 'Only Nav' means both communication and direction signs are disabled. 'Comm' means communication is enabled and 'Sign' means direction signs are enabled. 'Know=X' means that at the beginning of the simulation, the probability that an agent has prior knowledge of the environment is X. When omitted, Know=0.5. **a** 50 agents. **b** 100 agents. **c** 200 agents (Color figure online)

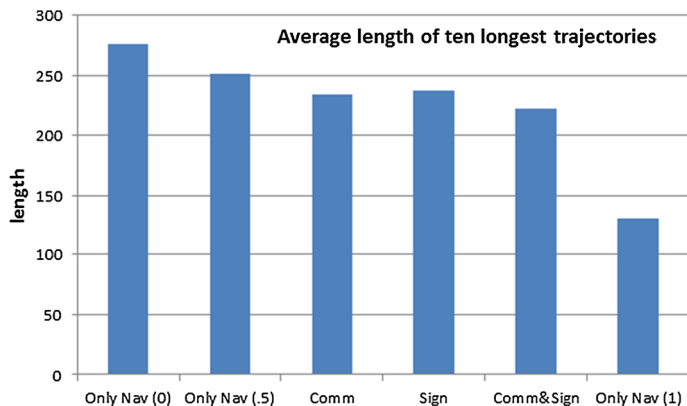


Fig. 10 The average lengths for the ten longest trajectories in each case. 'Only Nav' means both communication and direction signs are disabled. 'Comm' means communication is enabled and 'Sign' means direction signs are enabled. The values in parentheses represent the probability for an agent to have prior environment knowledge. This value is 0.5 when not given

compared to corresponding no communication (*Only Nav (0.5)* and *Sign*) ones. Together with the average evacuation times, these average lengths show that agents travel less but their evacuation times get worse when the communication is enabled.

Table 2 Some performance measurements with different number of agents in the evacuation scenario to show the computational cost of agent architecture (navigation, perception, and communication together)

| #agents | Min frame time (sec) | Max frame time (sec)* | Avg. frame rate (fps) |
|---------|----------------------|-----------------------|-----------------------|
| 0 | .016558 | .325687 | 27.42 |
| 50 | .016566 | .333333 | 25.12 |
| 100 | .016559 | .333333 | 21.61 |
| 200 | .026999 | .333333 | 13.53 |

The first row with no agents is given as a baseline and to emphasize the rendering cost of the environment. The communication is enabled but direction signs are disabled (*Comm* setting) in these simulations. (*) There is a maximum frame (delta) time limit employed in Unity game engine to prevent freezing and put a bound for frame rate. 0.333333 values correspond to this limit which ensures 3 fps. In between frame calculations are sometimes automatically interrupted and delayed to the next frame to achieve this rate

Table 2 summarizes some performance measurements in the evacuation scenario for different number of agents. These simulations were carried out on a PC with an Intel Xeon E5-2643 3.3GHz processor, 48GB RAM, and a Nvidia Quadro 4000 graphics processor. Minimum and maximum frame times are the smallest and largest values for the time between two consecutive frames. The last column shows the number of frames per second (fps) averaged over the simulation run. An important observation is that rendering (drawing) is often responsible for over 50% of all the computation (the exact value depends on what is in the camera view). The data for a simulation with no agents is given in the table so as to show the contribution of the rendering cost. Without any agents (i.e. no perception, navigation, or communication calculations), the average frame rate is 27.42 fps. The rest of the values in the table should be judged with respect to this baseline value.

4.3 Chat scenario

This final scenario is used to compare the trajectories and behaviors generated by our algorithm with those generated from a real-world scenario. We used a chat video (Fig. 11a) from crowd video data collected by the Movement Research Lab at Seoul National University [21]. Movement trajectories of the individuals were extracted from the video. Furthermore, we generated a 3D environment model in Unity that is similar to the environment observed in the video (Fig. 11b). Agents were placed at positions corresponding to the initial positions in the video. Simulations were run both with and without communication capability. Agents choose a random target as the goal position. Once they reach their target, they choose a new random target and this random selection continues until the simulation comes to an end. Agent movement trajectories from the simulations were extracted by recording agents' positions every second.

Vfractal estimation [26] is used to compare the real and simulated trajectories. 100 different step sizes are used varying from 0.1 to 10 with 0.1 increments. The length unit here is the size of a video pixel. All simulation coordinates, and thus, lengths, are scaled to match the unit. The scaling parameters (for both coordinates) are calculated by comparing the coordinates of corresponding positions on the floor geometry.

We applied the same vfractal calculations to (1) the trajectories extracted from the real video (blue), (2) the simulated trajectories without communication (red), and (3) the simulated trajectories where communication is enabled (green). The results of the calculations are summarized in Fig. 12. In these results, we observe that for both the estimations (solid

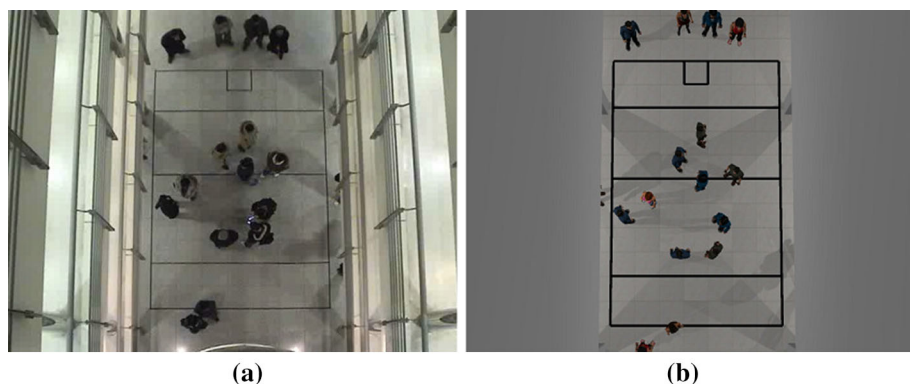


Fig. 11 Screenshots from the real video (a) and virtual simulation scene (b)

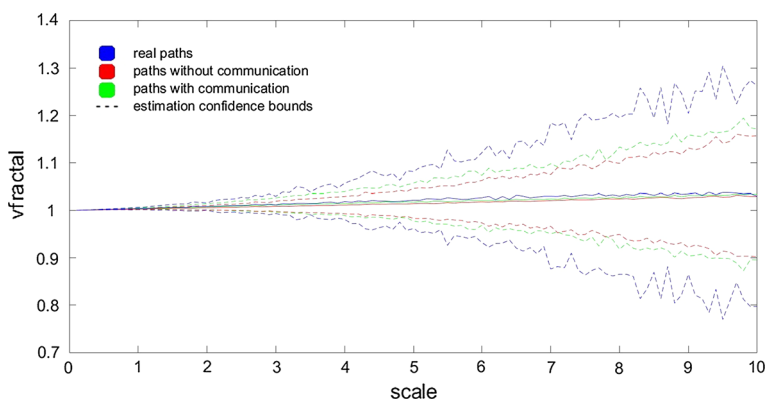


Fig. 12 Vfractal estimations and confidence bounds for trajectories from real video (blue), simulated trajectories without communication (red), and simulated trajectories with communication (green). Vertical axis is the estimated d value (fractal dimension) and horizontal axis is the scale (step size) used in estimation. Solid lines are the estimations and dashed lines with the same color are the confidence bounds in that estimation (Color figure online)

Table 3 Mean vfractal estimations and confidences calculated by averaging values in Fig. 12 over the range of different step sizes

| Trajectory | Avg. Vfractal | Confidence |
|--------------------|---------------|--------------|
| Real | 1.0209 | ± 0.0871 |
| Simulated (Comm) | 1.0178 | ± 0.0521 |
| Simulated (NoComm) | 1.0150 | ± 0.0421 |

lines) and the confidence in these estimations (dashed lines), the green values are closer to the blue values than the red values are to the blue values. The direct interpretation of this observation is that when communication is enabled, straightness/crookedness of simulated agent movement trajectories better match that of real trajectories. To further summarize the results, we calculate the averages of vfractal values and confidences over different step sizes. These averages are given in Table 3.

4.4 Analysis

It is possible to pinpoint some plausible autonomous behaviors when watching the simulation outputs (cf. the accompanying video). First of all, observed behavioral variety is improved as agents not only walk around but also autonomously engage in conversation. Even though our model only considers communication between two agents at a time, multiple instances of such communications occasionally happen at the same time, which can sometimes look like autonomous formation of standing-and-talking groups. This is consistent with the recent understanding that people in a crowd mostly move as a group rather than as individuals [25]. The measurements from the evacuation simulation, where ACMICS is applied to enable meaningful autonomous inter-agent communication, show that the change in behavior is consistent with obvious expectations about the effects on evacuation times and trajectories. The vfractal results in the last scenario show that when communication is enabled, the straightness/crookedness of the simulated trajectories are more in line with the real trajectories.

The components of the communication model other than the perception component are simple enough that they contribute little to the computational cost. The perception sub-components function like collision detection methods and have observable effects on the computation time. We believe that by improving and optimizing the perception implementation (e.g. instead of continuous tracking, perceived objects can only be checked when necessary) and by applying techniques such as space partitioning or level-of-detail, interactive rates can be achieved for higher number of agents.

5 Limitations, conclusions, and future work

Plausible autonomous behavior is a major aim in virtual crowd simulation research and it cannot be denied that communication takes place and affects behavior in the real-world crowds. We present a model for deliberate inter-agent information exchange in virtual crowds and investigate its effects on virtual crowd behavior. Our approach does not take into account subtleties of human behaviors and languages. Also, concepts of communication and information are considered at an abstract level, not in specific forms such as speech or sentences.

Our communication model, ACMICS, which uses a message structure based on the FIPA ACL message structure, is implemented as a PDA. Combined with the perception and navigation components, it was employed in four example scenarios. The accompanying video provides an overview, shows example visual outputs from simulations, and emphasizes important issues.

It should be noted that an important aim in our work was to develop a communication mechanism that can be easily employed in new scenarios. The low-level message type and scenario-independent tasks (AV layer) are separated from those that are high-level and scenario-dependent (FoE Layer). Using ACMICS in a new scenario only requires defining scenario-specific message types and high-level behavior related to sending and receiving specific messages.

Apart from visual simulation outputs, some measurements on flow rates, evacuation times, trajectory lengths, and trajectory shapes were presented and discussed. Visual analysis shows that observed behavioral variety is improved and examples of plausible behaviors such as autonomous grouping occur. The flow rates from the passageway scenario lead to two conclusions: (i) navigation mechanisms used behave similarly to the existing systems and (ii) communication does not cause a significant change in the passageway flow.

In order to simulate communication that influences behavior, we used an evacuation scenario. In this scenario, a separate algorithm was developed to model the evacuation behavior for an agent that does not have a priori knowledge about the environment. The average evacuation times acquired from simulating this scenario are consistent with the expectations. The use of direction signs improves average evacuation times but inter-agent communication increases them. When we looked at the lengths of the longest trajectories, however, there was a decrease with inter-agent communication. In other words, agents traveled less but took more time to evacuate when asking for and answering about direction was simulated through ACMICS.

The vfractal estimations in the last scenario were used to compare simulated trajectories with real trajectories extracted from a video. The results show that when agents communicate, simulation trajectories are closer to real trajectories in terms of straightness/crookedness.

The additional cost of simulating communication is a minor addition to the overall cost of multi-agent simulation. Further, most of this cost originates from the model of perception used, which can be improved. As it is a separate component in our architecture, changes made will not affect the communication model. There is also room for improvement through space partitioning and/or level-of-detail techniques.

It would be possible to better judge the realism of video outputs by performing user studies or quantitatively by developing similarity metrics for comparing crowd behavior. A more efficient perception model, an intermediate level of communication tasks regarding management of dialogs, combining the communication model with more sophisticated agent AI and applying it in new scenarios, including full-body crowd simulation and other interactions such as gaze [28] and face-to-face interactions [36], are other possibilities for future extensions, some of which we plan to investigate.

Acknowledgements This work was supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant No. 112E110. Additionally, the first author was supported by a scholarship (support type 2214-A) by TÜBİTAK to visit the University of North Carolina at Chapel Hill. We would like to thank Sarah George from the University of North Carolina-Chapel Hill for proofreading the paper.

References

1. Ali, S., Nishino, K., Manocha, D., & Shah, M. (Eds.). (2013). *Modeling, simulation and visual analysis of crowds, the international series in video computing* (Vol. 11). New York: Springer-Verlag.
2. Van den Berg, J., Lin, M., & Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, (pp. 1928–1935).
3. Berlo, D. K. (1960). *The process of communication: An introduction to theory and practice*. New York: Holt, Rinehart and Winston.
4. Blue, V., & Adler, J. (1999). Cellular automata microsimulation of bidirectional pedestrian flows. *Transportation Research Record: Journal of the Transportation Research Board*, 1678, 135–141.
5. Cassell, J., Sullivan, J., Prevost, S., & Churchill, E. F. (2000). *Embodied conversational agents*. Cambridge: MIT Press.
6. Chandler, D. (1994). The transmission model of communication. Online short paper at <http://users.aber.ac.uk/dgc/Documents/short/trans.html>. Accessed 24 Oct 2016.
7. Craig, R. (1999). Communication theory as a field. *Communication Theory*, 9, 119–161.
8. Curtis, S., Best, A., & Manocha, D. (2014). *Menge: A modular framework for simulating crowd movement*. Technical report: Department of Computer Science, University of North Carolina-Chapel Hill.
9. Durupinar, F., Güdükbay, U., Aman, A., & Badler, N. I. (2016). Psychological parameters for crowd simulation: From audiences to mobs. *IEEE Transactions on Visualization and Computer Graphics*, 22(9), 2145–2159.

10. Durupinar, F., Pelechano, N., Allbeck, J. M., Gbay, U., & Badler, N. I. (2011). How the Ocean personality model affects the perception of crowds. *IEEE Computer Graphics and Applications*, 31(3), 22–31.
11. Funge, J., Tu, X., & Terzopoulos, D. (1999). Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In *Proceedings of SIGGRAPH*, pp. 29–38.
12. Guy, S.J., Chhugani, J., Curtis, S., Dubey, P., Lin, M., & Manocha, D. (2010). Pedestrians: a least-effort approach to crowd simulation. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics symposium on computer animation*, pp. 119–128.
13. Guy, S.J., Kim, S., Lin, M.C., & Manocha, D. (2011). Simulating heterogeneous crowd behaviors using personality trait theory. In *Symposium on computer animation*, ACM, (pp. 43–52).
14. Harding, P., Gwynne, S., & Amos, M. (2011). Mutual information for the detection of crush. *PLOS One*, 6(12), 1–10.
15. Helbing, D., & Molnar, P. (1995). Social force model for pedestrian dynamics. *Physical Review E*, 51(5), 4282.
16. Henderson, L. (1974). On the fluid mechanics of human crowd motion. *Transportation Research*, 8(6), 509–515.
17. Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2007). *Introduction to automata theory, languages, and computation* (3rd ed.). Boston, MA: Pearson/Addison Wesley.
18. Integrated Environmental Solutions Ltd.: Simulex. <https://www.iesve.com/software/ve-for-engineers/module/Simulex/480>. Accessed 24 Oct 2016.
19. Kim, S., Guy, S.J., Manocha, D., & Lin, M.C. (2012). Interactive simulation of dynamic crowd behaviors using general adaptation syndrome theory. In *Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games*, ACM, (pp. 55–62).
20. Kullu, K., & Gbay, U. (2014). A layered communication model for agents in virtual crowds. In *Proceedings of 27th international conference on computer animation and social agents (CASA 2014)*, Short Papers. Houston, USA.
21. Lee, K.H., Choi, M.G., Hong, Q., & Lee, J. (2007). Group behavior from video: A data-driven approach to crowd simulation. In *Proceedings of the ACM SIGGRAPH/eurographics symposium on computer animation*, Eurographics Association, (pp. 109–118).
22. Mandelbrot, B. B. (1967). How long is the coast of Britain. *Science*, 156(3775), 636–638.
23. McDonnell, R., Larkin, M., Dobbyn, S., Collins, S., & O’Sullivan, C. (2008). Clone attack! perception of crowd variety. *ACM Transactions on Graphics*, 27(3), 26:1–26:8.
24. McDonnell, R., Newell, F., & O’Sullivan, C. (2007). Smooth movers: Perceptually guided human motion simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, Eurographics Association, (pp. 259–269).
25. Moussad, M., Perozo, N., Garnier, S., Helbing, D., & Theraulaz, G. (2010). The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLoS One*, 5(4), 1–7.
26. Nams, V. O. (1996). The vfactal: A new estimator for fractal dimension of animal movement paths. *Landscape Ecology*, 11(5), 289–297.
27. Nara, A., & Torrens, P.M. (2007). Spatial and temporal analysis of pedestrian egress behavior and efficiency. In *Proceedings of the 15th annual ACM international symposium on advances in geographic information systems*, ACM, New York, NY, USA (pp. 59:1–59:4).
28. Narang, S., Best, A., Randhavane, T., Shapiro, A., & Manocha, D. (2016). PedVR: Simulating gaze-based interactions between a real user and virtual crowds. In *Proceedings of the 22nd ACM conference on virtual reality software and technology*, ACM, New York, NY, USA (pp. 91–100).
29. Nwana, H. S. (1996). Software agents: An overview. *The Knowledge Engineering Review*, 11(03), 205–244.
30. Pan, X. (2006). *Computational modeling of human and social behaviors for emergency egress analysis*. Ph.D. thesis, The Department of Civil and Environmental Engineering, Stanford University.
31. Park, S. I., Quek, F., & Cao, Y. (2013). Simulating and animating social dynamics: Embedding small pedestrian groups in crowds. *Computer Animation and Virtual Worlds*, 24, 155–164.
32. Pelechano, N. (2006). *Modeling realistic high density autonomous agent crowd movement: social forces, communication, roles and psychological influences*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.
33. Pelechano, N., Allbeck, J.M., & Badler, N.I. (2007). Controlling individual agents in high-density crowd simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation* (pp. 99–108). Eurographics Association.
34. Pelechano, N., Allbeck, J.M., & Badler, N.I. (2008). Virtual crowds: Methods, simulation, and control. Synthesis Lectures on computer graphics and animation #8. Morgan & Claypool Publishers

35. Poslad, S. (2007). Specifying protocols for multi-agent systems interaction. *ACM Transactions on Autonomous and Adaptive Systems*. doi:[10.1145/1293731.1293735](https://doi.org/10.1145/1293731.1293735).
36. Randhavane, T., Bera, A., & Manocha, D. (2016). *F2FCrowds: Planning agent movements to enable face-to-face interactions*. Technical report: Department of Computer Science, University of North Carolina-Chapel Hill.
37. Schramm, W. (1997). How communication works, chap. 3, pp. 51–63. Greenwood Publishing Group (1954). (Reprint in) *Mass Media and Society* by A. Wells, ed.
38. Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language* (Vol. 626). Cambridge: Cambridge University Press.
39. Shannon, C. E., & Weaver, W. (1949). *The mathematical theory of communication*. Champaign: University of Illinois Press.
40. Shoulson, A., Marshak, N., Kapadia, M., & Badler, N.I. (2013). ADAPT: the agent development and prototyping testbed. In *Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games* (pp. 9–18). ACM.
41. Silverman, B. G., Bharathy, G., & Cornwell, K. O. J. (2006). Human behavior models for agents in simulators and games: Part II: Gamebot engineering with PMFserv. *Presence: Teleoperators and Virtual Environments*, 15, 163–185.
42. Silverman, B. G., Johns, M., Cornwell, J., & O'Brien, K. (2006). Human behavior models for agents in simulators and games: Part I: Enabling science with PMFserv. *Presence: Teleoperators and Virtual Environments*, 15, 139–162.
43. Snook, G. (2000). Simplified 3D movement and pathfinding using navigation meshes. In M. DeLoura (ed.), *Game programming gems* (pp. 288–304). Newton Center, MA: Charles River Media.
44. Stevenson, A. (Ed.). (2010). *Oxford Dictionary of English* (3rd ed.). Oxford: Oxford University Press.
45. Sun, L., Shoulson, A., Huang, P., Nelson, N., Qin, W., Nenkova, A., et al. (2012). Animating synthetic dyadic conversations with variations based on context and agent attributes. *Computer Animation and Virtual Worlds*, 9, 17–32.
46. Thalmann, D. (2006). Populating virtual environments with crowds. In *Proceedings of the ACM international conference on virtual reality continuum and its applications* (pp. 11–11). ACM, New York, NY, USA.
47. Thalmann, D., & Musse, S. R. (2013). *Crowd Simulation* (2nd ed.). London: Springer-Verlag.
48. Torrens, P. M., Nara, A., Li, X., Zhu, H., Griffin, W. A., & Brown, S. B. (2012). An extensible simulation environment and movement metrics for testing walking behavior in agent-based models. *Computers, Environment and Urban Systems*, 36(1), 1–17.
49. Unity Technologies: Unity®. <http://unity3d.com/>. Accessed 24 Oct 2016.
50. Watzlawick, P., Bavelas, J. B., Jackson, D. D., & O'Hanlon, B. (2011). *Pragmatics of Human communication: A study of interactional patterns, pathologies and paradoxes*. New York: W. W. Norton.
51. Yu, Q., Terzopoulos, D. (2007). A decision network framework for the behavioral animation of virtual humans. In *Symposium on computer animation* (pp. 119–128).