Towards Animating Virtual Humans in Flooded Environments

Diogo Schaffer
diogo.schaffer@acad.pucrs.br
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Rio Grande do Sul, Brazil

André Antonitsch
andre.antonitsch@acad.pucrs.br
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Rio Grande do Sul, Brazil

Amyr Neto
amyrneto@gmail.com
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Rio Grande do Sul, Brazil

Soraia Musse
soraia.musse@pucrs.br
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Rio Grande do Sul, Brazil

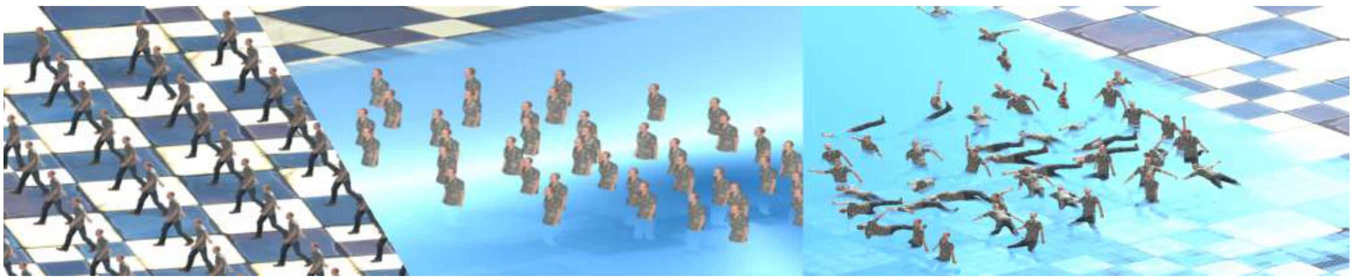


Figure 1: On the left: virtual humans are walking in the environment, without fluids, going to their goals; in the center: virtual humans can still walk to their goals in a weak fluid stream and; on the right: virtual humans cannot walk anymore towards their goals, once they are carried by the fluid.

ABSTRACT

The simulation of virtual humans organized in groups and crowds has been widely explored in the literature. Nevertheless, the simulation of virtual humans that interact with fluids is still incipient. Indeed it is easy to understand that human behavior is different from ordinary rigid bodies when affected by fluids, i.e., on the one hand, agents can try to walk, achieve their goals against fluid forces, trying to survive. On the other hand, humans can also be completely carried by the fluid, depending on the conditions, as a passive rigid body. A challenge in this area is that virtual agent simulation research often focuses on the realism of their trajectories and interaction with the environment, obstacles, and other agents, without considering that agents might evolve into an environment that can take control of their movements and trajectories in certain conditions. In this case, it is essential to note that, with proper integration between agents and fluids, we should be able to simulate agents who can continue walking despite an existing fluid (e.g., a weak fluid stream), walking with an effort to stay in the desired direction (e.g., medium stream), until they are partially or totally carried by a fluid, like a strong flow of water in a river or the sea. The main contribution of our model is to give the first step into simulating the steering behaviors of humans in environments with fluids. We integrate two published methodologies and available source codes in order to create our method. For the motion of virtual humans, we use BioCrowds; and SPlisHSPlasH as a fluid dynamics model. Results indicate that the proposed approach generates coherent behaviors regarding the influence of fluids on people in real events, even if this is not the objective of this paper, because other variables should be incorporated, in cases of serious simulations.

MIG '20, October 16–18, 2020, Virtual Event, SC, USA

ACM ISBN 978-1-4503-8171-0/20/10…$15.00
https://doi.org/10.1145/3424636.3426900

CCS CONCEPTS

• **Computing methodologies** → **Animation**; **Computer graphics**.

KEYWORDS

flow of fluids, groups of virtual humans, flooded environments

ACM Reference Format:
Diogo Schaffer, André Antonitsch, Amyr Neto, and Soraia Musse. 2020. Towards Animating Virtual Humans in Flooded Environments. In *Motion, Interaction and Games (MIG '20), October 16–18, 2020, Virtual Event, SC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3424636.3426900

1 INTRODUCTION

In the last decades, many algorithms have been proposed to simulate behavior and motion of groups of virtual humans and crowds [Helbing and Molnar 1998; Van Den Berg et al. 2011]. In general, virtual humans (VH) into a crowd are simulated as positions in the environment, and their motion use navigation algorithms to achieve predefined goals while avoiding collisions with obstacles and other agents [Pelechano et al. 2017; Thalmann and Musse 2013]. Also, as strategy to mimic real crowds, work has been done acquiring real crowd data, and using this data to simulate crowd behaviour [Musse et al. 2007]. Furthermore, work has been done towards adding realism to crowd simulation, including animation features which increase the perception of a realistic crowd [Kapadia et al. 2013; Tecchia et al. 2001]. Finally, there are work in the literature considering external forces, such as pushing and collisions [Kim et al. 2013].

Our objective here is to give a first step towards simulating virtual humans under influence of environmental forces, such as fluid forces of nature (i.e. wind and water streams, although here we focus on liquids). One situation that has not been explored in the literature and applications, such as games, regards the motion of individuals in terrains with dry to wet or flooded ground variation and how such material impacts the individuals' motion as illustrated in Figure 1. The challenge here is that moving on dry grounds can be completely different from moving on flooded ones. In this paper, we propose the integration of groups of VH with a model of Smoothed Particle Hydrodynamics (SPH), which is considered as one of the key concepts for fluid animation in Computer Graphics [Gissler et al. 2019]. We present a novel model where VH movements can be affected by fluids at various levels, i.e., from no fluid interference until not being able to walk anymore and being completely dragged by the fluid. In the mid-term VHs should have both algorithms working together to generate coherent motion. In order to provide more realism to agents' behavior, we also propose agents endowed with different abilities to fight for their lives, i.e., some variable which represents the agents' strength or vulnerability against fluid dynamics.

Concerning fluid dynamics, Dynamic simulation of mechanical effects has a long history in Computer Graphics. Physical simulation using the Eulerian grid-based approach was proposed by Foster and Metaxas [Foster and Metaxas 1996], and they were the first to propose solving the full 3D Navier Stokes equations to simulate the visual properties of dynamic fluids. Later, Foster and Fedkiw [Foster and Fedkiw 2001] proposed an extension of such a technique to simulate liquids using particles inside it. In particular, Smoothed Particle Hydrodynamics (SPH) is considered as one of the key concepts for fluid animation in Computer Graphics [Bender et al. 2019; Cornelis et al. 2019; Gissler et al. 2019; Koschier and Bender 2017]. Therefore, in the last decades, many have been achieved in Physical simulation, including rigid and deformable bodies [Akinci et al. 2012; Koschier and Bender 2017; Macklin et al. 2014].

The main contribution of this work is to endow virtual humans with the possibility to be controlled in a hybrid way, i.e., by crowd microscopic control being affected by fluids and, at the same time reacting individually to BioCrowds stimuli, where both interfere in the agents' motion. To achieve this goal, we propose to integrate a crowd simulation model with an SPH based fluid dynamics simulation model. We use BioCrowds [de Lima Bicho et al. 2012] as the model to simulate steering behaviors and collision avoidance of groups of agents and SPlisHSPlasH [Gissler et al. 2019] to perform SPH simulation.

2 RELATED WORK

In the present work, we aim for a unified SPH and agent steering model, so both areas in the literature are of relevance. Models to simulate liquids using SPH approach exists since pioneer work [Gingold and Monaghan 1977; Lucy 1977]. In this area, work has been done to solve problems regarding boundary handling [Bender et al. 2019; Koschier and Bender 2017], viscosity [Takahashi et al. 2015; Weiler et al. 2018], drag forces [Macklin et al. 2014] and multi-phase fluid simulation [Solenthaler and Pajarola 2008]. The approach presented by Keiser et al [Keiser et al. 2006] introduce variation of coarseness based on resolution needs, i.e, the particles are bigger (coarse) in regions where there is not much interaction of the fluid, and they use smaller (grained) particles to increase fluid resolution in regions that are interacting with objects, or the fluid surface. Finally, there are works on improvement of pressure solvers, which can be seen as the fluid simulation itself [Cornelis et al. 2019; Raveendran et al. 2011]. Besides the ability to properly simulate fluid behavior based on smoothed particles, work has been developed to deliver proper rendering [Bender et al. 2017; Orthmann et al. 2013; Xiao et al. 2017] and to achieve real-time simulation results using the computational power of modern GPUs [da Silva et al. 2010; Mokos et al. 2015; Tafuni et al. 2018; Xia and Liang 2016]. A general survey on SPH based fluid simulation can be found in the state-of-the-art report of Ihmsen et al. [Ihmsen et al. 2014].

Regarding the steering models literature, since pioneer work of Reynolds [Reynolds 1987], and Helbing [Helbing and Molnar 1998], a lot has been done. Solutions based on competition for free space [de Lima Bicho et al. 2012], scripted behaviour [Thalmann and Musse 2013], velocity and geometric approaches [Van Den Berg et al. 2011] were proposed. More recently, a method to improve the agents' individualism was proposed by Pelechano et al. [Pelechano et al. 2017] aiming to simulate more realistic heterogeneous crowds.

On the simulation of humans interacting with fluids, the work by Yang, Laszlo & Singh [Yang et al. 2004] aims to animate a virtual character in a swimming environment. The approach is based on the physical simulation of a dynamic human figure model in a simplified fluid model. Yet in the field of animating humans in fluids, the work by Bermudez et al. [Bermudez et al. 2018] integrates drag forces to promote realistic movement of a character interacting with fluids from existing motion data. The work by Carensac et al. [Carensac et al. 2015] animates partially immersed character gait using a controller based on a simple hydrodynamics model that produces natural looking movement adapting to variations on liquid depth, density and viscosity.

For the present work, we elected the BioCrowds model [de Lima Bicho et al. 2012], since it is a collision-free method and freely available under Unity3D platform upon request. For the SPH model, we elected the SPlisHSPlasH Framework [Gissler et al. 2019], which implements state of the art fluid compression solvers and provides source code for practical integration with the BioCrowds model. Both methods are briefly depicted herein.

2.1 The BioCrowds Model

BioCrowds method [de Lima Bicho et al. 2012] proposes the environment discretization with uniformly distributed markers. Agents in the environment compete for those markers, based on proximity criteria, and use them to determine their movement vectors. Indeed, each agent i accesses the markers inside its personal space R_i to search for markers that are closest to i than any other agent j. So, a marker is only available to the closest agent. For a given agent i, with a set of N available markers denoted by $S = \{\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_N}\}$, we calculate its movement vector $\mathbf{m}$ using Equation 1:

$$\mathbf{m} = \sum_{k=1}^{N} w_k(\mathbf{a_k} - \mathbf{X}), \tag{1}$$

where $\mathbf{a_k}$ is the marker's position and $\mathbf{X}$ is the agent's position. The marker's weight w_k is calculated from Equation 2:

$$w_k = \frac{f(\mathbf{g} - \mathbf{X}, \mathbf{a_k} - \mathbf{X})}{\sum_{l=1}^{N} f(\mathbf{g} - \mathbf{X}, \mathbf{a_l} - \mathbf{X})}, \tag{2}$$

where $\mathbf{g}$ is the goal position of agent i. Function f should prioritize markers that lead the agent directly to its goal: a possible choice is defined in Equation 3:

$$f(x, y) = \frac{1 + cos\theta}{1 + ||y||}, \tag{3}$$

where θ is the angle between x and y. The model should allow the agent to move with a maximum desired speed S_{max}. However, in dense crowds, the space available for each agent is smaller, resulting in a speed reduction. Therefore, in the proposed model, the instantaneous motion vector can be defined by:

$$v = s_{min} \frac{\mathbf{m}}{||\mathbf{m}||}, \tag{4}$$

where $s_{min} = min(||\mathbf{m}||, S_{max})$ which implies that if $||\mathbf{m}|| > S_{max}$, the maximum displacement is limited by S_{max}. Otherwise, it is given by $||\mathbf{m}||$. Please refer to BioCrowds original paper [de Lima Bicho et al. 2012] for further details about the method.

2.2 The SPlisHSPlasH Model

Akinci et al. [Akinci et al. 2012] proposed a Generic Iterative Pressure Solver with boundary handling. In their model, a predicted fluid velocity $\mathbf{v}_f^{*,l}$ is initialized with non-pressure induced velocities. Then, fluid particles velocities are refined at each iteration (l) of their algorithm. Velocities at rigid particles $\mathbf{v}_r^*$ are predicted the same way, but not computed every iteration as fluid velocities. According to Gissler et al. [Gissler et al. 2019], the lack of refinement of rigid particles velocities during the iterations may introduce errors. To address this issue, Gissler and colleagues propose an improved two-way coupling where all forces are applied to both fluid particles and rigid particles in each iteration, resulting in more precise fluid particles and rigid particles velocities, since predicted rigid velocities influence the pressure refinement and thus affecting fluid velocities.

To cope with SPH simulation in our work, we elected the method proposed by Gissler et. al [Gissler et al. 2019], which presents a robust fluid-rigid coupling for Smoothed Particle Hydrodynamics (SPH) fluids and rigid bodies with particle-sampled surfaces. The work provides an implementation of pressure solvers, which are used in this work to resolve hydrodynamics and provide with data on forces involved in a scenario of liquid flow. Among the pressure solvers implemented by The SPlisHSPlasH Framework, we used Divergence-free smoothed particle hydrodynamics (DFSPH) to simulate SPH in our experiments. Since the mass and velocity of fluid impact inelastic collisions, we generate scenarios with different volumes and velocities of fluid, thus varying fluid flow strengths.

To avoid unstable behavior, we do not create instances of agents in the SPlisHSPlasH simulation. Due to forces inherent to agents' movements, opposing particles' movement, instability may occur on collisions, leading to undesired behavior. Furthermore, our goal is to study agents' movements from the steering point of view. For this purpose, a resultant velocity vector of the particles colliding each agent, called $\mathbf{v}_F$ from now on, is computed from the total fluid mass and velocity. This approach can be made from any fluid dynamics simulation model capable of providing such data.

3 MODEL OVERVIEW

We first address the problems coupling BioCrowds to SPlisHSPlasH in Section 3.1, where we explain the changes made in BioCrowds to cope with the fluid stream. In Section 3.2, we describe new features of Fluid Endurance included in BioCrowds.

3.1 Connecting to SPlisHSPlasH Model

The integration of BioCrowds with SPlisHSPlasH is done via memory mapping. An array of floats is shared in memory for the SPlisHSPlasH implementation to write particles' data (velocity and position) every frame. In Unity3D, the BioCrowds model reads this memory space and updates instances of particles with their mass, velocities, and positions for both visualization and simulation purposes. Finally, some controllers are also shared in memory to keep both BioCrowds and SPlisHSPlasH simulations in sync.

Particles' positions projected to the ground i.e., $Y = 0$ are used as the center of circular regions, with radius = 0.25 for particles. This seems coarse at first glance, but work has been done with coarse fluid particles, such as the work of Keiser and colleagues [Keiser et al. 2006], which consider coarse fluid particles, where resolution is not an issue; and more grained particles in object boundaries for precision of object interaction. Since this first attempt to give crowd agents interaction with fluids is meant for games, efficiency is an issue. Furthermore, we have tested our model with ten times the number of particles, them being proportionally smaller than presented here. The results of those experiments showed not much significantly changes in our quantitative results, although the performance dropped significantly. Finally, our agents are not impacting particles movements, so we decided to trade particles resolution for lower computational costs.

The agents in our model have a radius of 0.5. The circular regions formed by agents' and particles' radius are used to determine collisions, i.e., when $d < r1 + r2$, where d is the distance between the two bodies (particle and agent) and $r1$ and $r2$ are the respective radius. We optimize this search by considering only particles in the vicinity of the agent to test for collisions, i.e., particles which are in the neighbouring cells in the discretized environment[1], where cells have size 2x2m. Particles' velocities and masses are then used to compute inelastic collisions [Beer et al. 2013] against agents. These collisions, in turn, impact agents' trajectories. The resultant vector

[1]We use the same space discretization as proposed by Bicho et al. [de Lima Bicho et al. 2012]

of fluid particle's momenta denoted as $\mathbf{p}_{F,a}$ is computed at each frame t, for all particles that collide with each agent a, according to Equation 5.

$$\mathbf{p}_{F,a}^{t} = \sum_{i} m_i \mathbf{v}_{F,i}^{t}, \tag{5}$$

where m_i is the mass and $\mathbf{v}_{F,i}^{t}$ is the velocity of every i particle from the simulated fluid in collision with agent a, at frame t. Also, the resultant colliding particles' mass $m_{c,a}$ is computed by the sum of each particle i colliding agent a: $m_{c,a} = \sum_i m_i$.

In an analogous way, agent's a momenta at frame t, when simulated in the context of BioCrowds, states for $\mathbf{p}_{B,a}^{t} = m_a \mathbf{v}_{B,a}^{t}$, where agents' mass m_a is an input parameter, and $\mathbf{v}_{B,a}^{t}$ is computed according to Equation 4.

The particles' momenta $\mathbf{p}_{F,a}^{t}$ and mass m_c are then used to impact the velocity of agent a at frame $t+1$. The impact of fluid over the agent a is computed as an inelastic collision given by Equations 6, 7 and 8, according to Newton's law of conservative motion [Beer et al. 2013], which changes agents' velocities according to their masses and desired velocities $\mathbf{v}_{B,a}$. The conservation of total momentum of the two bodies can be expressed as:

$$\mathbf{p}_{B,a}^{t} + \mathbf{p}_{F,a}^{t} = \mathbf{p}_{B,a}^{t+1} + \mathbf{p}_{F,a}^{t+1}, \tag{6}$$

where $\mathbf{p}_{*}^{t}$ are the bodies momenta before collision (time t), and $\mathbf{p}_{*}^{t+1}$ are the bodies momenta after collision (time $t+1$). For an inelastic collision, the final velocity of the two bodies are the same. We can denote this final velocity as $\mathbf{v}_{a}^{t+1}$, and so we can rewrite Equation 6 as:

$$(m_a + m_c)\mathbf{v}_{a}^{t+1} = \mathbf{p}_{B,a}^{t} + \mathbf{p}_{F,a}^{t}, \tag{7}$$

and the terms can be rearranged to find the new agent's velocity, computed as:

$$\mathbf{v}_{a}^{t+1} = \frac{\mathbf{p}_{B,a}^{t} + \mathbf{p}_{F,a}^{t}}{m_a + m_c}. \tag{8}$$

It is important to notice here that we do not perform any change in the particle's velocity, since this is an one-way approach. For future work, we will consider the presence of agents to change particle's velocity and thus changing the dynamics of fluids as well.

3.2 Flooding BioCrowds Agents

In addition to providing an integration with SPlisHSPlasH Model, the present work improves BioCrowds agents by including a new feature: a skill to endure against the effects of fluid streams. In real-life, people can offer various resistances to certain fluids, depending on their own weight, ability to swim, height, and etc.

According to described in [Wallingford 2006], the result of the scenario is impacted by people's vulnerability, the area where the situation occurs, and flood hazard inherent to fluid flow strength. So, we propose an individual variable that aims to represent people's vulnerability. Each agent is provided with a skill representing its endurance against fluid streams effects. This endurance is noted by τ and is used as attenuation of the fluid influence over agent a. The parameter τ_a lies in the range $[0, 1]$, being *zero* no influence from the fluid over the agent, and 1 the fluid impacts the agent at its full strength. As mentioned before, it can map many variants of this scenario such as the individual ability to move in water streams, the individual strength, height, weight, wearing of gear (such as fins), emotional state, friction against the floor, or any other factor that might help the agent on fighting against the stream. One should want to break τ in other parameters, such as separating environment related and agent related characteristics, and equate them to compute τ (keeping the range $[0, 1]$). But, in this work, for the sake of simplicity, we keep it as one single parameter serving as wildcard for every aspect. Furthermore, one might want to vary τ for simulating dynamics in the scenario, such as fatigue. To apply the effects of τ_a in the computation of agent's speed, we changed Equation 8 and rewrote it as Equation 9.

$$\mathbf{v}_{a}^{t+1} = \frac{\mathbf{p}_{B,a}^{t} + \tau_a \mathbf{p}_{F,a}^{t}}{m_a + \tau_a m_c}. \tag{9}$$

If $\tau_a = 0.0$, it would nullify the particles' momenta (and mass) impacting agent a, so agents move only according to BioCrowds, which can maybe lead to an unrealistic behavior (i.e., agents ignore the fluid since the second terms of the equation are equal to zero). On the other hand, if $\tau_a = 1.0$, the fluid fully impacts agent a motion. Experimental results in Section 4.2 shows the impact of different τ values on agents' trajectories.

As our goal is to provide a method for animation, we have included the possibility of interactively change the τ values in a particular frame, for a specific agent or a group. Figure 2 illustrates an example of animation control, where the vulnerability of agents (τ) changes from 0.8 to 0.0 (from highly vulnerable to not vulnerable at all) at frame 870 (28 seconds from the beginning) in (c). Besides, in (d) we can see that agents move toward to their goal, according to BioCrowds.

4 EXPERIMENTAL RESULTS

In this section we describe the tested scenarios, the variation on parameters used to evaluate our model, and then we comment on the obtained results. The scenarios are detailed in Section 4.1 along with the parameters chosen for evaluation. The results obtained in the performed simulations are presented in Section 4.2 while Section 4.3 further explores our model's capabilities on a situation where a group of agents evolves against the fluid stream, and another situation with a fixed obstacle present in the scenario. Finally, in Section 4.4, we provide a discussion showing the coherence of our methodology with real-life data.

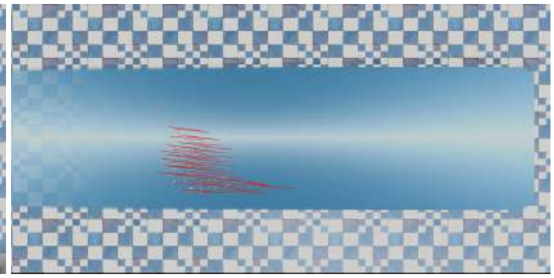
4.1 Simulation Scenarios

To evaluate our model, we define as a scenario to be tested a fluid stream modeling a river 27m wide, 1m deep and 100m long in an environment 50m wide and 100m long. When particles from SPlisHSPlasH simulator reach the end of the scenario (left side of the box), they are recycled to keep fluid streaming. We used a rectangular particle emitter that can generate a maximum of 100,000 particles in the scene[2]. The particles move from right to left in images.

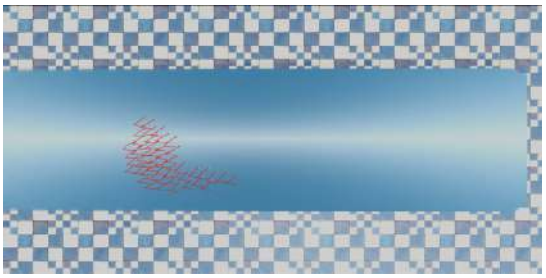
This scenario is designed to evaluate changes in trajectories of 50 agents trying to cross a river stream with the desired speed of 1.3m/s. All agents start on one side of the river (in the bottom of images),

[2]Basically, we used default parameters to run SPlisHSPlasH as defined in file Emitter.json, changing only the fluid velocity ($\mathbf{v}_F$).

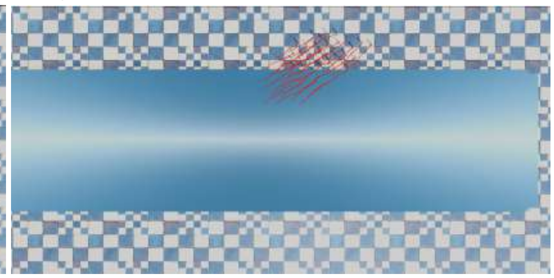
(a) $\tau = 0.8$ and $\mathbf{v}_F = 2.5$ at time 6s.



(b) $\tau = 0.8$ and $\mathbf{v}_F = 2.5$ at time 28s. At this instant, τ changes to $\tau = 0.0$.



(c) $\tau = 0.0$ and $\mathbf{v}_F = 2.5$ at time 32s. The effects of the change are noticeable.



(d) $\tau = 0.0$ and $\mathbf{v}_F = 2.5$ at time 70s. Agents move to their goals.0

Figure 2: Four different frames of a simulation where τ changes from 0.8 to 0.0 at frame 870 (approximately time 28 seconds). In (a) and (b) we see agents being carried by the fluid, in (c) the abrupt change in trajectories occurs when $\tau = 0.0$, few seconds earlier and in (d) the agents are achieving their goals. Fluid is blue, agents trajectories are red.

and their goals are placed on the opposite side, on top (Figure 3). Agents that are carried by the fluid stream and reach the left side of the box are considered dead agents. Agents that achieve their goals are considered survivors. As illustrated in Figure 4, all agents are animated and rendered as virtual humans on Unity. Agents' red traces represents their trajectories, and the fluid is represented by transparent-blue squared particles.

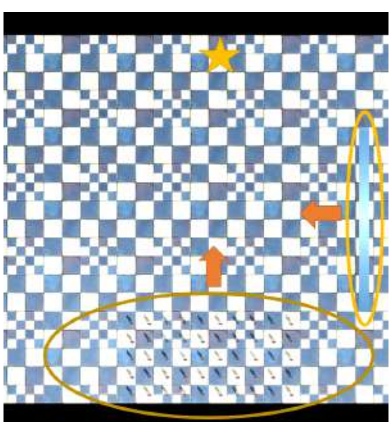


Figure 3: Illustration of simulated environment. On the bottom we can see the 50 agents, on the right the fluid starting to evolve, from right to left, and on top agent's goal is illustrated as a star.

The baseline scenario is obtained when $\tau = 0$, i.e., agents do not take into account the fluid as a factor to impact their trajectories. Simulations S1, S2 and S3 were executed changing velocities and keeping $\tau = 0$ in order to obtain the baseline data (S1/S2/S3 in the first line of Table 1). As mentioned before, on higher values of τ, and also in most hazardous environments (i.e., those with most stronger fluid streams), it may turn impossible for people to keep control of their trajectories, according to the work of Viseu [VISEU 2006]. In such hazardous situations, we provide a feature to animate characters as ragdolls, with only visualization purposes, so that agents look like they are being dragged away by a mighty river, instead of just walking downstream. We perform this animation in a simplistic manner, by placing the hip of the 3D character in a height equivalent to the mean height of all colliding particles. Then we rotate agent's hip around all axis, by a rate according to the particles speed, increased by a random number in the range $[0.1, 0.5]$. Figure 4 (right-bottom) illustrates an image with those animations. In our experiments, ragdoll animation is enabled when $\tau \geq 0.8$ and initial fluid velocity $\mathbf{v}_F > 2$. We decided to use those values simply based on empirical observation, i.e. those were the conditions which agents potentially will not achieve their goals.

In the presented simulations we considered that agent's mass is fixed (=65kg) and varied attributes: τ and fluid velocity ($\mathbf{v}_F$) according to presented in Table 1. Results show the time when the last agent achieved the goal and we also present the difference (%Diff) w.r.t. baseline time (S1, S2 and S3). In the case agents are carried by the fluid all the way to the left side of the box, they cannot achieve the goal at any time during the simulation, and so we wrote NA (not achieved) in the table. In those cases, no agent survived the experiment.

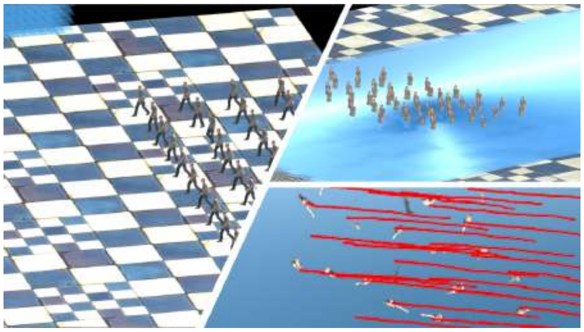


Figure 4: On the left we can see the agents walking on the floor, on the right (top) agents keep walking due to the weak fluid and on the bottom, it is possible to see the agents being animated as ragdolls in Unity, and their trajectories in red.

Table 1: Planned simulation scenarios: these scenarios vary input τ and $\mathbf{v}_F$ and keep mass of agents fixed in $m_a = 65kg$. S1, S2 and S3, in bold, represent the baseline. Time to goal specifies time the last agent reaches the goal, or not (NA), and %Diff describes the difference between the baseline time and the analysed simulation. In S6, 15 agents reach the left side of the box and 35 achieved their goals.

Simulation	Variants			
ID	τ_a	$\mathbf{v}_F(m/s)$	Time to goal (s)	% Diff
S1/S2/S3	0.0	0.5/1.5/2.5	46	-
S4	0.2	0.5	54	18%
S5	0.2	1.5	93	102%
S6	0.2	2.5	160	247%
S7	0.8	0.5	97	110%
S8	0.8	1.5	NA	-
S9	0.8	2.5	NA	-
S10	1.0	0.5	140	204%
S11	1.0	1.5	NA	-
S12	1.0	2.5	NA	-

After evaluating the impact of the fluid and agents' τ values on the trajectories, we experimented our model in two different situations with agents moving against the stream and obstacles. The scenario remains basically the same in both new experiments, but in one situation we tested two groups of agents: one group, similar to the previous experiment, attempts to cross the river, while the other group attempting to move against the fluid stream, evolving from left to right. The other scenario experimented with the presence of an obstacle. We instantiated a fixed obstacle in both BioCrowds and SPlisHSPlasH, in such a way that the obstacle interacts with both the fluid and the agents. The results of these two experiments are exposed in Section 4.3.

4.2 Simulation Results

This section discusses the impact of fluid on the agents' trajectories. Starting with baseline experiment, Figure 5 illustrates 50 agents that are not impacted by the fluid. This happens because parameter τ is set to zero.

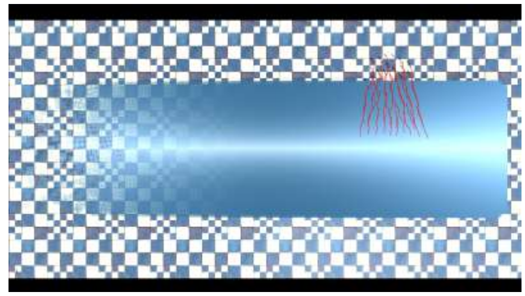


Figure 5: Agents trajectories ignoring the fluid. Same trajectories emerge independently of $\mathbf{V}_F$.

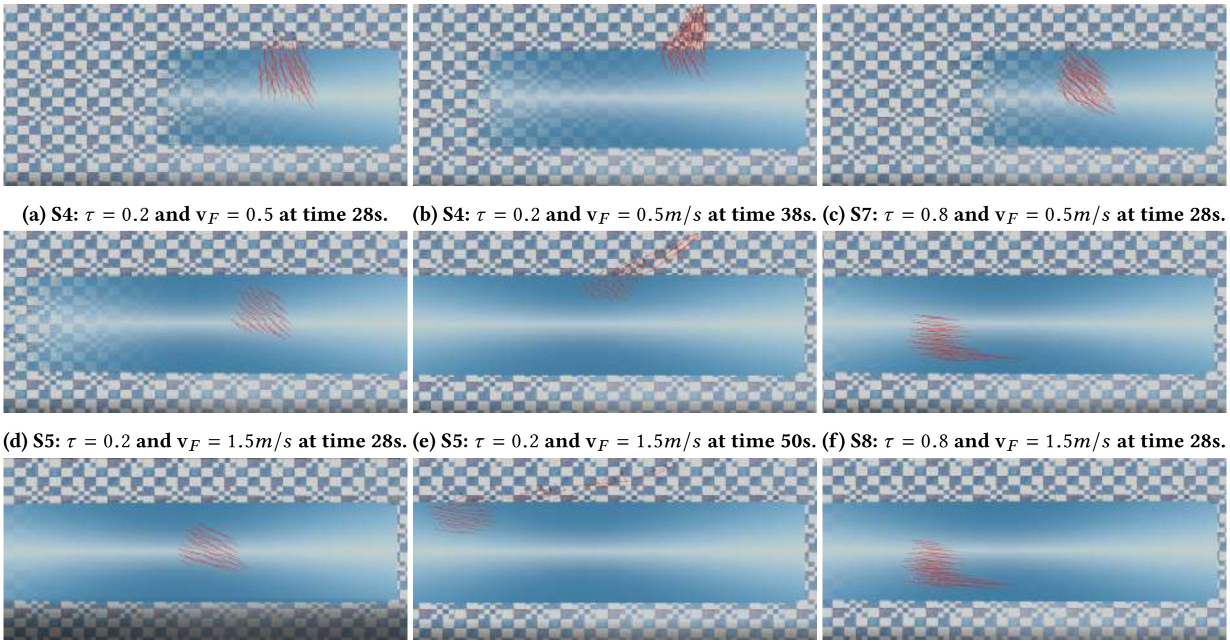
To show the impact of τ and fluid velocity variation, we depicted a set of frames from simulations S4 to S12 (see Table 1) in Figure 6. The three figures on the top row shows scenarios with $\mathbf{v}_F = 0.5m/s$, while the middle and bottom rows presents $\mathbf{v}_F = 1.5m/s$ and $\mathbf{v}_F = 2.5m/s$, respectively. This way, it is possible to compare the different trajectories resultant from different velocities. The left and middle columns of Figure 6 shows an evolution in time of simulations S4, S5, and S6 (times on figure's caption), with $\tau = 0.2$. The rightmost column of Figure 6 (c, f, and i) depicts a single frame from the remaining S7, S8, and S9 experiments, respectively, with $\tau = 0.8$. This column shows the variation of velocity in a higher τ situation. As mentioned before, the agents' goal is on the top, their traces are red, and fluid evolves from right to left. Also, remind that the agents' mass is kept constant ($m_a = 65kg$) for all experiments.

The weaker fluid stream tested is set to $\mathbf{v}_F = 0.5m/s$ and, as stated before, shown in the top row of Figure 6 (a, b and c) depicting simulations S4 and S7. The middle and bottom rows show a faster fluid stream keeping τ and visualized frames. Notice that, in all cases, the higher the velocity, the greater is the deviation of agents' trajectories: they are carried a bit further to the left. This deviation impacts their success or failure in reaching their goals. Agents fail to reach their goals when they lose control of their own trajectories, being dragged away by the fluid stream. In Figure 6h, for instance, 15 agents are dragged away by the river stream and are unable to reach their goals while the other 35 agents get to their goals, thus saving themselves.

According to Equation 9, the smaller is the value of τ_a the stronger is the agent a to endure against fluid forces or, in other words, it means that the fluid has less impact on agent's motion. To evaluate the impact of τ value in the obtained results, one must compare the left- and rightmost columns of Figure 6, where the same simulation time (or frame) is being displayed. It is possible to note that on the rightmost column of Figure 6 agents are more impacted by the fluid strength, compared to the leftmost column. The visual results are coherent with which would be expected, regarding the effects of increasing velocity and agents' vulnerability on final agents' motion [Wallingford 2006].

4.3 Other Scenarios

In this section, we present other tested scenarios. In the first case, we define a scenario with two groups in the same river crossing situation as before, except now there is another group of agents moving against the fluid stream. Fifty (50) agents must cross from



(a) S4: $\tau = 0.2$ **and** $\mathbf{v}_F = 0.5$ **at time 28s.** **(b) S4:** $\tau = 0.2$ **and** $\mathbf{v}_F = 0.5m/s$ **at time 38s.** **(c) S7:** $\tau = 0.8$ **and** $\mathbf{v}_F = 0.5m/s$ **at time 28s.**

(d) S5: $\tau = 0.2$ **and** $\mathbf{v}_F = 1.5m/s$ **at time 28s.** **(e) S5:** $\tau = 0.2$ **and** $\mathbf{v}_F = 1.5m/s$ **at time 50s.** **(f) S8:** $\tau = 0.8$ **and** $\mathbf{v}_F = 1.5m/s$ **at time 28s.**

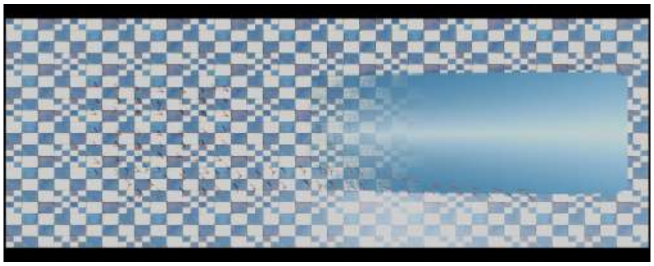
(g) S6: $\tau = 0.2$ **and** $\mathbf{v}_F = 2.5m/s$ **at time 28s.** **(h) S6:** $\tau = 0.2$ **and** $\mathbf{v}_F = 2.5m/s$ **at time 70s.** **(i) S9:** $\tau = 0.8$ **and** $\mathbf{v}_F = 2.5m/s$ **at time 28s.**

Figure 6: Simulation frames from S4 to S9 with different fluid velocities (0.5, 1.5 and 2.5m/s), $\tau = 0.2$ **and** 0.8 **values; and** $m = 65kg$**. It is possible to see that as higher is the** $\mathbf{v}_F$ **(g), (h) and (i) the longer are the agents trajectories in order to reach their goals. It is also possible to see that as higher is the** τ **value, more the fluid impacted the agent motion, for example comparing (a) with (c), (d) and (f) and (g) and (i). Figures (b), (e) and (h) illustrate the evolution of S4 in (a), S5 in (d) and S6 in (g).**

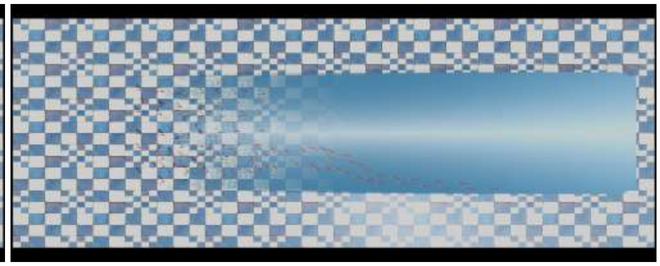
the bottom to the top in Figure 7a (as in previous scenarios), while other 50 agents move against the fluid direction (from left to right). As before, we evaluate the change in trajectories of agents, due to fluid interaction, setting parameter $\tau = 0.6$ and fluid velocity ($\mathbf{v}_F = 1.5m/s$), which configures a Very High danger situation (Table 2).

Figure 7 illustrates some frames of this experiment, picturing the evolution of agents during the simulation. Figure 7a shows the simulation at time=10s, when the fluid has just reached some agents from the bottom. Figures 7b, 7c and 7d show the progression of the agents at times 15s, 20s and 30 seconds after the beginning of the simulation respectively. Figure 7a shows agents in the bottom beginning to get carried by the fluid, while agents on the left are still evolving to the right. At time 15s (Figure 7b), the fluid begins to oppose left agents' movement and, meanwhile, some agents from the bottom start colliding with the ones from the left. At time 20s, in Figure 7c, most agents are being carried by the fluid, including agents from the left, which are now being dragged back to their starting point: they are unable to beat the stream. Finally, in Figure 7d (at 30s) all agents are being carried by the fluid and soon will get to the extreme left, where they fail to achieve their goal. Actually, in this experiment, only 8 agents succeed in achieving the goal. This would be expected in a scenario classified as very dangerous, according to [VISEU 2006].

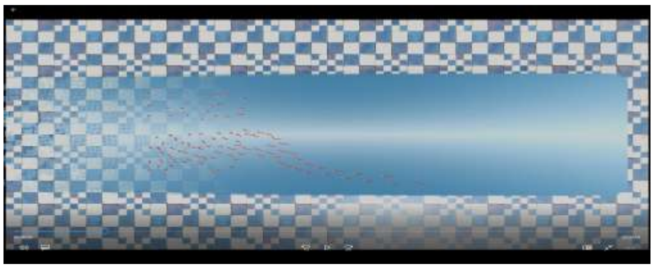
We also experimented with an obstacle instantiated both in BioCrowds and SPlisHSPlasH, so that both agents and fluid interacts with them. This creates turbulent regions and vortices in the fluid, which in turn impacts agents' trajectories in a different manner. Figure 8 illustrates the scenario with an obstacle $7m \times 7m$ and $1m$ high. In Figure 8a it is pictured the initial condition of these experiments, at time=2s, showing 52 agents moving from the bottom to the top of the scenario. The experiment is run with fluid velocity $\mathbf{v}_F = 1.5m/s$ and $\tau = 0.6$. Same as before, the fluid evolves from the right to the left represented as transparent blue squares. The obstacle is pictured in dark grey. Figure 8b shows simulation time at $15s$, when some agents are protected from the fluid behind the obstacle while others are still being carried by the fluid. At this instant, the obstacle gives shelter to agents creating a safe zone behind it. Notice also the fluid passing above the obstacle. This happens because the fluid pressure forces particles to move around and above the obstacle, as simulated by SPlisHSPlasH. Figure 8c shows simulation at time=$25s$, depicting some agents coming out on the other side of the obstacle and being carried by the fluid, while agents behind the obstacle are still safe from the fluid, but the safe zone is about to end. As we can see in the Figure 8d (time=30s), the safe zone is now flooded, and all agents are being carried by the river stream, although some are almost done crossing. After 57s, in Figure 8e, some agents start moving out of the water. Those agents are now safe, since there is nothing impairing them from achieving their goals: it is just a matter of time now. At time 1min20s, some more agents move out of the water, while many are still being dragged. In fact, for this experiment, 26 agents succeed in achieving their goals (against 8 for the same conditions without obstacles, in last scenario). In fact, the obstacle creates shelter for the agents and helps them crossing the river.



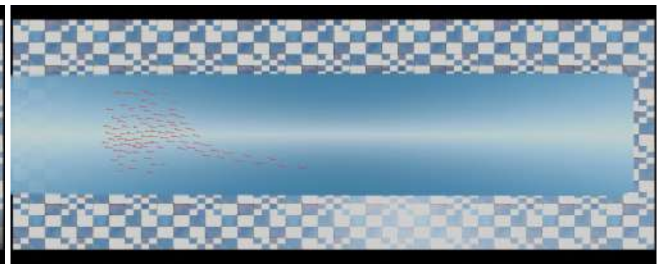
(a) Time 10s: Agents on the bottom begin to enter the fluid and their trajectories start to change.



(b) Time 15s: Agents from the bottom start being carried and their trajectories collide with the other group.



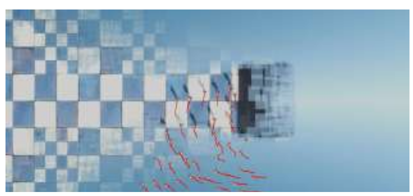
(c) Time 20s: Almost all agents are being carried by the fluid. Agents moving against the fluid begin to lose control of their trajectories and are being pushed back.



(d) Time 30s: Almost all agents are being carried by the fluid, and soon major part of them will not be able to get to their goals anymore. Actually, only 8 agents save themselves in this experiment.

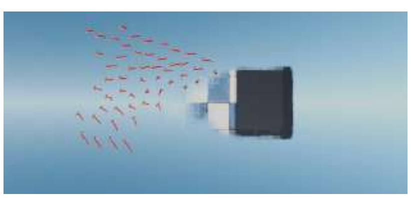
Figure 7: Simulation scenario of two groups of agents. One has 50 agents moving from the bottom to the top, while the other group has 50 agents moving against the fluid stream. The experiment is performed with $\tau = 0.6$ and fluid velocity $\mathbf{v}_F = 1.5m/s$. In Figure 7a it is possible to see both groups of agents in the bottom and left. Only 8 agents achieve their goals.

(a) The setup of our scenario with an obstacle ($\tau = 0.6$, $\mathbf{v}_F = 1.5m/s$, at time $2s$), pictured in dark grey. On the bottom, 52 agents move to the top.



(b) Picture at time=15s from the beginning of the simulation. Some agents hide from the fluid behind the obstacle (Zoomed image).

(c) At time=25s, some agents are coming out the other side of the obstacle, and the fluid is catching them again. Also, it begins to flood behind the obstacle (Zoomed image).



(d) Agents pass the safe zone at time=30s. They are all being carried by the fluid (Zoomed image).

(e) Some agents begin to exit the fluid stream at time=57s.

(f) At time=1m20s many agents start to leave the fluid and others fail to achieve the goal (26 agents achieve their goal).

Figure 8: Scenario with a $7m \times 7m$ obstacle ($1m$ high) instantiated both in BioCrowds and SPlisHSPlasH.

4.4 Results Evaluation

As seen in the last section, parameters τ from the agents and $\mathbf{v}_F$ from the fluid impact the motion of agents in different levels, changing the time agents achieve (or not) their goals, as presented in Table 1. Therefore, we compare our results with the impact a fluid should have on real humans. According to Viseu [VISEU 2006], there are four levels of danger to human beings depending on the fluid velocity, denoted as $\mathbf{v}_F$, and the stream height or, in our case scenario, river depth denoted as H, as resumed in Table 2. According to this table, when $H \times \mathbf{v}_F > 1.0$, it is expected that people struggle against the stream, being this range classified as Very High danger level. This means that, at this point, most people would get dragged away by the stream, being unable to reach the other side. Although this appears to be true, according to our results, the study of Teresa Viseu [VISEU 2006] evaluates the risks of inhabiting river banks and downstream of dams from the civil engineering point of view, and was pointed to us by specialists as consolidated reference on the matter. In her work, she evaluates the risks involved on floods, rupture of dams and by the power of river streams. The evaluation of risks, made in her work, results in four levels of risk, in function of water velocity and height of water flood, as depicted in Table 2.

Table 2: Levels of danger to humans [VISEU 2006]

Danger level	**Dynamic flood** ($H \times v_F$)
Low	$H \times \mathbf{v_F} < 0.5m^2/s$
Moderate	$0.5m^2/s < H \times \mathbf{v_F} < 0.75m^2/s$
High	$0.75m^2/s < H \times \mathbf{v_F} < 1.0m^2/s$
Very High	$H \times \mathbf{v_F} > 1.0m^2/s$

Using the insights of Viseu's work, we attempt to compare our model to real world. This effort does not long to prove exactitude, because our model do not aim to cope with all real-world variables, once we want to provide a version for games. Instead, we intend to establish a base to compare our model and evaluate its realism from a fair perspective. So, considering that Viseu's assessment of danger classifies as very high danger floods which are faster than $1.0m^2/s$, and reminding that our flood height is $H = 1.0m$, this means that if our velocity $\mathbf{v}_F$ is higher than $1.0m/s$, we have a very high danger situation. Indeed, by comparing our results from Table 1, in our experiments, whenever $\mathbf{v}_F > 1.0m/s$, no agent could achieve their goals, except when the simulated agents were powerful, i.e., $\tau = 0.2$. And even though agents are powerful ($\tau = 0.2$) in simulation S6 with $\mathbf{v}_F = 2.5m/s$, 15 out of 50 agents still fail to achieve their goals. Yet, on Table 1, no agents reach their goals when $\tau = 0.8$ or $= 1.0$, and $\mathbf{v}_F > 1.0$.

We make notice that our river length is 100m, and for the purpose of evaluating the likelihood to reality, we consider in our model that there is a fall by the end (leftmost side of screen). Agents who get pushed beyond this point fall on it, and are considered unable to reach their goals anymore. If the river had been longer, the number of such failing agents would be different, making our results dependant of our scenario choice. Finally, there are many aspects in real life which are not being considered in this work such as the possibility of people getting injured or tired, entering in panic and also the presence of debris in the fluid stream, which can be harmful. If such limitations are disregarded, and we focus on the fluid velocity, according to Table 2 our obtained results show coherence if compared to real-life.

In addition, Table1 also presents in the last column the difference among the simulations time and the baseline. We can see that differences seem coherent, i.e., time to reach goals increase as $\mathbf{v}_F$ and τ increases too, although we do not have data to provide an accurate comparison with real-life.

For the scenario with agents moving against the stream, we could see that most part of them are unable to endure against it, and are pushed backwards. At the same time, the other group of agents trying to cross the river collide against them. As discussed before, this configures Very High danger situations (Table 2) and it is expected that agents fail on fighting against the stream. In fact, according to Figure 7d, 42 out of 50 agents fail to achieve their goals in this simulation run.

Finally, the scenario with obstacle showed some heterogeneity on the fluid flow distribution of our scenario. This happens because the cubic obstacle creates turbulence in the fluid particles as they collide against it. This resulted in a protected area behind the obstacle, where agents evolve almost undisturbed by the fluid. At this point they just move toward their goals. But, as they come out on the other side of the obstacle, the stream of fluid caches them again. On the experiment with an obstacle, 26 out of 50 agents achieved their goals with success. The obstacle, in fact, helped the agents by protecting a portion of their way against the fluid forces.

5 FINAL CONSIDERATIONS

This paper presents a model to integrate character motions and fluid dynamics [Gissler et al. 2019] to provide coherent animation of characters impacted by the environment. We propose a way for agents to take into account the simulated particles together with the steering behaviors provided by BioCrowds [de Lima Bicho et al. 2012]. The experiments show that fluid streams considerably impact agents' trajectories. Furthermore, the stronger the fluid, the more drag is shown by the trajectories, up to a point where agents are no longer able to reach the other side. By endowing agents with a vulnerability skill against the fluid, we could obtain a tool that can serve for animation, but in the same time is coherent with real-life, as discussed by Viseu [VISEU 2006].

We performed experiments with fixed fluid parameters to focus on character behaviors, but certainly, other parameters can be tested. In the same way, we use only 50 to 100 agents, but more agents can be considered once BioCrowds main focus is crowd simulation. Also, other group distribution, making agents pursue different goal positions is an addition. Same goes for a greater number of obstacles to simulate small villages or group of buildings. This configuration may occur in more chaotic fluid streams, and thus impacting agent's trajectories in various ways, such as higher fluid pressures at narrow places, or presence of safe zones behind walls.

For future work, we aim to create two-way simulation, making the presence of agents interact with the fluid stream. The integration of this feature might create turbulence in the fluid flow and, in turn, change the impact of fluid in agents' trajectories. As in the scenario with an obstacle, agents can be obstacles themselves, and dense crowds may (or may not) block the stream to a point where they help other agents saving themselves. This brings up the possibility to create connections between agents, as if they hold hands, and scenarios where agents save each other may be tested, including rescuer agents.

Another possible future contribution is to escalate these small scenarios to huge scenarios, enabling the simulation of entire cities and the neighboring area. This ability can make possible the simulation of dam breaks, such as the one occurred on the January 25^{th}, 2019 in Brumadinho [Raman and Liu 2019], Minas Gerais, Brazil. Simulation tools may help avoid such disasters, aid in emergency plans to minimize damage, and, in the occurrence of such a catastrophe, the same simulation tools may help find victims quickly.

ACKNOWLEDGMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. The authors would like to thank funding agencies CNPq and FAPERGS. We also would like to thank Prof. Jaime Gomes for his directions on fluid behaviour and the pointing of the work by Teresa Viseu [VISEU 2006] which enlightened our insights about the problem of dam breaks and its outcomes.

REFERENCES

Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. 2012. Versatile rigid-fluid coupling for incompressible SPH. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 62.

Ferdinand P. Beer, E. Russell Johnston Jr., David F. Mazurek, Phillip J. Cornwell, and Brian P. Self. 2013. *Vector Mechanics for Engineers: Statics and Dynamics.* McGraw-Hill Companies, Inc.

Jan Bender, Dan Koschier, Tassilo Kugelstadt, and Marcel Weiler. 2017. A Micropolar Material Model for Turbulent SPH Fluids. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '17)*. ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3099564.3099578

Jan Bender, Tassilo Kugelstadt, Marcel Weiler, and Dan Koschier. 2019. Volume Maps: An Implicit Boundary Representation for SPH. In *Motion, Interaction and Games* (Newcastle upon Tyne, United Kingdom) *(MIG '19)*. ACM, New York, NY, USA, Article 26, 10 pages. https://doi.org/10.1145/3359566.3360077

Luis Bermudez, Jerry Tessendorf, Daniel Zimmermann, and Victor Zordan. 2018. Real-Time Locomotion with Character-Fluid Interactions *(MIG '18)*.

Samuel Carensac, Nicolas Pronost, and Saida Bouakaz. 2015. Real-Time Gait Control for Partially Immersed Bipeds *(MIG '15)*.

Jens Cornelis, Jan Bender, Christoph Gissler, Markus Ihmsen, and Matthias Teschner. 2019. An Optimized Source Term Formulation for Incompressible SPH. *Vis. Comput.* 35, 4 (April 2019), 579–590. https://doi.org/10.1007/s00371-018-1488-8

Jose Ricardo da Silva, Jr., Esteban W. Clua, Paulo A. Pagliosa, and Anselmo Montenegro. 2010. Fluid Simulation with Rigid Body Triangle Accuracy Collision Using an Heterogeneous GPU/CPU Hardware System. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (Washington, D.C.) *(I3D '10)*. ACM, New York, NY, USA, Article 20, 1 pages. https://doi.org/10.1145/1730804.1730992

Alessandro de Lima Bicho, Rafael Araújo Rodrigues, Soraia Raupp Musse, Cláudio Rosito Jung, Marcelo Paravisi, and Léo Pini Magalh aes. 2012. Simulating crowds based on a space colonization algorithm. *Computers & Graphics* 36, 2 (2012), 70 – 79. https://doi.org/10.1016/j.cag.2011.12.004 Virtual Reality in Brazil 2011.

Nick Foster and Ronald Fedkiw. 2001. Practical Animation of Liquids. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. ACM, New York, NY, USA, 23–30. https://doi.org/10.1145/383259.383261

Nick Foster and Dimitri Metaxas. 1996. Realistic Animation of Liquids. *Graph. Models Image Process.* 58, 5 (Sept. 1996), 471–483. https://doi.org/10.1006/gmip.1996.0039

R. A. Gingold and J. J. Monaghan. 1977. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 181, 3 (12 1977), 375–389. https://doi.org/10.1093/mnras/181.3.375 arXiv:http://oup.prod.sis.lan/mnras/article-pdf/181/3/375/3104055/mnras181-0375.pdf

Christoph Gissler, Andreas Peer, Stefan Band, Jan Bender, and Matthias Teschner. 2019. Interlinked SPH Pressure Solvers for Strong Fluid-Rigid Coupling. *ACM Trans. Graph.* 38, 1, Article 5 (Jan. 2019), 13 pages. https://doi.org/10.1145/3284980

Dirk Helbing and Peter Molnar. 1998. Social Force Model for Pedestrian Dynamics. *Physical Review E* 51 (05 1998). https://doi.org/10.1103/PhysRevE.51.4282

Markus Ihmsen, Jens Orthmann, Barbara Solenthaler, Andreas Kolb, and Matthias Teschner. 2014. SPH Fluids in Computer Graphics. In *Eurographics 2014 - State of the Art Reports*, Sylvain Lefebvre and Michela Spagnuolo (Eds.). The Eurographics Association. https://doi.org/10.2312/egst.20141034

Mubbasir Kapadia, Alexander Shoulson, Funda Durupinar, and Norman I. Badler. 2013. Authoring Multi-actor Behaviors in Crowds with Diverse Personalities. In *Modeling, Simulation and Visual Analysis of Crowds - A Multidisciplinary Perspective*, Saad Ali, Ko Nishino, Dinesh Manocha, and Mubarak Shah (Eds.). The International Series in Video Computing, Vol. 11. Springer, 147–180. https://doi.org/10.1007/978-1-4614-8483-7_7

Richard Keiser, Bart Adams, Philip Dutré, Leonidas Guibas, and Mark Pauly. 2006. *Multiresolution particle-based fluids.* Technical Report 520. Department of Computer Science, ETH Zurich. 10 pages.

Sujeong Kim, Stephen Guy, and Dinesh Manocha. 2013. Velocity-based modeling of physical interactions in multi-agent simulations. *Proceedings - SCA 2013: 12th ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (07 2013). https://doi.org/10.1145/2485895.2485910

Dan Koschier and Jan Bender. 2017. Density Maps for Improved SPH Boundary Handling. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '17)*. ACM, New York, NY, USA, Article 1, 10 pages. https://doi.org/10.1145/3099564.3099565

Leon B Lucy. 1977. A numerical approach to the testing of the fission hypothesis. *The astronomical journal* 82 (1977), 1013–1024.

Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. 2014. Unified Particle Physics for Real-Time Applications. *ACM Trans. Graph.* 33, 4, Article 153 (July 2014), 12 pages. https://doi.org/10.1145/2601097.2601152

Athanasios Mokos, Benedict D. Rogers, Peter K. Stansby, and José M. Domínguez. 2015. Multi-phase SPH modelling of violent hydrodynamics on GPUs. *Computer Physics Communications* 196 (2015), 304 – 316. https://doi.org/10.1016/j.cpc.2015.06.020

Soraia R. Musse, Cláudio R. Jung, Julio C. S. Jacques Jr, and Adriana Braun. 2007. Using computer vision to simulate the motion of virtual agents. *Computer Animation and Virtual Worlds* 18, 2 (2007), 83–93. https://doi.org/10.1002/cav.163 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cav.163

Jens Orthmann, Hendrik Hochstetter, Julian Bader, Serkan Bayraktar, and Andreas Kolb. 2013. Consistent Surface Model for SPH-Based Fluid Transport. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Anaheim, California) *(SCA '13)*. Association for Computing Machinery, New York, NY, USA, 95–103. https://doi.org/10.1145/2485895.2485902

Nuria Pelechano, Jan M. Allbeck, Mubbasir Kapadia, and Norman I. Badler. 2017. *Simulating Heterogeneous Crowds with Interactive Behaviors.* Taylor and Francis Group.

Arun Raman and Fei Liu. 2019. An investigation of the Brumadinho Dam Break with HEC RAS simulation.

Karthik Raveendran, Chris Wojtan, and Greg Turk. 2011. Hybrid Smoothed Particle Hydrodynamics. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Vancouver, British Columbia, Canada) *(SCA '11)*. ACM, New York, NY, USA, 33–42. https://doi.org/10.1145/2019406.2019411

C. W. Reynolds. 1987. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH computer graphics* 21, 4 (July 1987), 25–34.

B. Solenthaler and R. Pajarola. 2008. Density Contrast SPH Interfaces. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Dublin, Ireland) *(SCA '08)*. Eurographics Association, Goslar, DEU, 211–218.

A. Tafuni, J.M. Domínguez, R. Vacondio, and A.J.C. Crespo. 2018. A versatile algorithm for the treatment of open boundary conditions in Smoothed particle hydrodynamics GPU models. *Computer Methods in Applied Mechanics and Engineering* 342 (2018), 604 – 624. https://doi.org/10.1016/j.cma.2018.08.004

Tetsuya Takahashi, Yoshinori Dobashi, Issei Fujishiro, Tomoyuki Nishita, and Ming Lin. 2015. Implicit Formulation for SPH-based Viscous Fluids. *Computer Graphics Forum* 34 (05 2015). https://doi.org/10.1111/cgf.12578

Franco Tecchia, Céline Loscos, Ruth Conroy, and Yiorgos Chrysanthou. 2001. Agent Behaviour Simulator (ABS): A Platform for Urban Behaviour Development. *GTEC'2001, January* (03 2001).

D. Thalmann and S. R. Musse. 2013. *Crowd Simulation.* Springer-Verlag.

Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. 2011. Reciprocal n-body collision avoidance. In *Robotics research.* Springer, 3–19.

Teresa VISEU. 2006. Valley safety downstream dams. Methodologies for risk management support. (In portuguese: Segurança dos vales a jusante de barragens. Metodologias para apoio à gestão do risco). *Lisboa: LNEC* (2006), 122. http://repositorio.lnec.pt:8080/xmlui/handle/123456789/15575?show=full

HR Wallingford. 2006. Flood Risks to People Phase 2. *Final Report Prepared for Defra/Environment Agency Flood and Coastal Defence R&D Programme* (2006). http://randd.defra.gov.uk/Document.aspx?Document=FD2321_3437_TRP.pdf

Marcel Weiler, Dan Koschier, Magnus Brand, and Jan Bender. 2018. A Physically Consistent Implicit Viscosity Solver for SPH Fluids. *Computer Graphics Forum* 37 (05 2018), 145–155. https://doi.org/10.1111/cgf.13349

Xilin Xia and Qiuhua Liang. 2016. A GPU-accelerated smoothed particle hydrodynamics (SPH) model for the shallow water equations. *Environmental Modelling and Software* 75 (2016), 28 – 43. https://doi.org/10.1016/j.envsoft.2015.10.002

Xiangyun Xiao, Shuai Zhang, and Xubo Yang. 2017. Real-Time High-Quality Surface Rendering for Large Scale Particle-Based Fluids. In *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (San Francisco, California) *(I3D '17)*. Association for Computing Machinery, New York, NY, USA, Article 12, 8 pages. https://doi.org/10.1145/3023368.3023377

Po-Feng Yang, Joe Laszlo, and Karan Singh. 2004. Layered Dynamic Control for Interactive Character Swimming. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Grenoble, France) *(SCA '04)*. Eurographics Association, Goslar, DEU, 39–47. https://doi.org/10.1145/1028523.1028529