Towards a Legion of Virtual Humans: Steering Behaviors and Organic Visualization

André da Silva Antonitsch
PUCRS
Porto Alegre, Brazil
andre.antonitsch@acad.pucrs.br

Soraia Raupp Musse
PUCRS
Porto Alegre, Brazil
soraia.musse@pucrs.br

Luiz Henrique de Figueiredo
IMPA
Rio de Janeiro, Brazil
lhf@impa.br

***Abstract*—Many studies have been done to control virtual humans and crowds. Navigation, split, merge, and collision avoidance are examples of what has been done in such areas when simulating crowds of individuals. This paper explores the macroscopic crowds' concept and seeks to solve two main issues: macroscopic crowd control and visualization. In particular, we are interested in providing steering behaviors applied to macroscopic models of crowds, in this case, called Legions of people, which are an abstraction for a vast amount of people. To provide the steering behaviors, we propose a multi-level control of such structures that can represent less or more people, in an emergent way. We also propose a new organic visualization of macroscopic and huge crowds based on blobby models.**

***Index Terms*—virtual human crowds, blobby models.**

I. Introduction

Many methods have been proposed in the literature to simulate crowds of virtual humans. Typically, agents in a crowd are simulated individually (in a microscopic way [1], [2], [3]), just like their applied behavioral and steering algorithms. On the other hand, macroscopic methods aim to simulate aggregated motion as flows of people [4], [5], generally without considering individual behaviors. While macroscopic crowds usually make it possible to simulate huge crowds, because it optimizes the needed computation, it brings some challenges for visualization and control [6], [7].

Here is an example of a control issue: when it comes to adapting to the environment's complexity, a microscopic crowd can split into smaller crowds (always simulated through individuals) to move in complex environments [8]. When faced with an obstacle, real people need to choose a path to go around it, left or right. In the same way, crowds of people, when faced with obstacles, can split into smaller groups that follow several different paths around the obstacles to avoid collisions. After splitting, if there is enough free space, groups can merge and form a new crowd structure. In macroscopic crowds, this kind of control needs split and merge behaviors in multi-level behaviors, i.e., subdividing large crowds into smaller crowds, once there are not simulated individuals.

The second issue regards visualization. It is known that microscopic simulations of crowds allow more realistic and organic simulation than macroscopic heat maps or people flow visualizations [9], [10].

In this paper, we seek to contribute with a new crowd abstraction model that solves the two problems mentioned above. First, to adapt the crowd's behavior according to the complexity of the virtual environment: crowds should split or merge, increasing or decreasing the number of structures, respectively. To that end, we provide a multi-level control, from macroscopic (huge crowds) to microscopic (individual agents), passing through groups of different sizes, in an emergent way. The main goal is to be efficient, as macroscopic control allows and simultaneously relies on the advantages of microscopic control, e.g., the more accurate behaviors necessary to adapt to the complexity of the environment. Our second contribution is in the visualization aspect. We propose a blobby [11] approach to visualize macroscopic crowds, producing more organic visuals for the simulated structures.

Our crowd abstraction, named *Legion*, is based on BioClouds, proposed by Antonitsch et al. [12]. However, our legions can split into smaller groups when moving towards an obstacle. The main difference between Legions and BioClouds is the multi-level control of crowds: we can deal with huge crowds, smaller ones, and groups until reaching one individual level. Legions can split and merge as they navigate the environment.

The paper is structured as follows: Section II introduces related work on macroscopic, microscopic, and hybrid crowd simulation; Section III describes our new models for legion splitting and merging; Section IV describes our new method for visualization; Section V presents our experimental results; and Section VI discusses our conclusions and future work.

II. Related Work

The field of crowd simulation is vast, and its applications can range from fire drill planning, crime prevention, and pedestrian flow analysis [9]. The analysis of how a crowd behaves in an environment as a whole is largely the main concern of the field. This section reviews microscopic models of crowd simulation, where each agent is modeled as a single entity of the simulation, and macroscopic models, where the model control is globally applied for after being instantiated to individuals. Microscopic crowd simulation models each agent in a crowd as a single entity of simulation. Examples of this model are the social forces model [1], ORCA [2], and BioCrowds [3].

Helbing et al. [1] introduced the concept of social forces. This model conceptualizes agent movement as if influenced by internal and environmental characteristics, such as desired velocity, distance from other pedestrians, and a term modeling attractive effects of simulation. These simulations exhibit some self-organizing behavior observed in real crowds.

Van den Berg et al. [2] proposed the ORCA model, which simulates a crowd based on reciprocal velocity obstacles. The simulation is solved with linear programming, each step taking the linear time. Created initially for robotics controllers, the model tends to give little personal space to each agent. The model has extensions to allow more natural crowd forming behavior, e.g., a simulation which respects the fundamental diagram [13] that describes how fast each given crowd is willing to move in a certain density, used for modeling cultural aspects, for instance.

Treuille et al. [6] proposed Continuum Crowds, a crowd simulation model based on continuum field dynamics. The model simulates a crowd as a dynamic potential field to perform navigation for agents and moving obstacles.

Xiong et al. [14] proposed a hybrid approach for crowd simulation, separating the simulation environment into microscopic and macroscopic regions. They proposed leveraging the accuracy and computational efficiency of both models. Their work uses the ORCA model as a microscopic simulation model, restricted to a region of interest; the remaining space is simulated using the continuum crowds model. The regions are interfaced with a transition region. When an agent crosses the boundary, it is removed from one simulation and added to the other. The regions are defined during the environment description and remain static during the simulation.

Bicho et al. [3] introduced a space discretization and competition model for crowd simulation. In their BioCrowds model, agents compete to maintain their personal space, and to do so will take possession of space around them. To model the environment, Bicho et al. [3] use space discretization markers. Agents take possession of markers closest to them, forming an emergent Voronoi partitioning of space. The convex Voronoi cells each agent now occupies guarantee collision avoidance, as long as no agent attempts to move outside its cell. BioCrowds also features self-organizing emergent crowd behavior, notably the emergence of lanes, vortices, bottleneck, and arc formation. BioClouds [12] is a generalization of BioCrowds [3] and aims to abstract a massive number of people as if it were a single entity. BioClouds inherits some properties of BioCrowds, like evolving in discrete space (dots on the floor called *markers* which are organized in a regular grid) and simulating the next positions based on the competition model (one agent can only possess markers on the floor, and no more), which causes free-of-collision movement. BioClouds models entities as groups of agents called clouds, which can be a group of just one individual, even or large crowds (no upper limit is imposed). A cloud's behavior differs from a BioCrowds agent in the following aspects: the space capturing radius and the distance function used for space competition. While in BioCrowds, the capture radius is static, in BioClouds it is dynamic and determined by the cloud desired density in this simple way: as the density of people$/m^2$ increases in a specific cloud, the perception radius increases and provides the possibility to occupy more space around it. Another difference between BioCrowds and BioClouds is the distance function. Clouds use the power-of-a-point distance function, which generates power diagrams, a generalization of Voronoi diagrams. Power diagrams display the mathematical properties necessary to the BioCrowds collision avoidance model. Once the agents (or clouds) spaces are correctly represented as Voronoi diagrams, the method is free-of-collision. As proposed by Antonitsch et al. [12], Clouds cannot be subdivided; only their radius can be increased or decreased. This fact causes issues when clouds evolve in complex environments. BioClouds have only two levels of simulation: clouds and agents.

III. Legions

We propose a new macroscopic crowd abstraction, called *Legion*, to address the issue of multi-level behavior, allowing the simulation of groups of varied sizes that react to the complexity of the environment. Legions represent crowds of people without simulating individuals, as proposed in BioClouds model [12]. We introduce the possibility of having different levels between the higher level (macroscopic) and the individual agent (microscopic) in an emergent way. The Legion can be subdivided into crowds, smaller crowds, large groups, smaller groups, and successively down to the level of individuals. The subdivision of vast legions into smaller groups of various sizes occurs depending on the environment.

The main contribution of our model is that it can deal with those varied sized groups as a single entity, keeping the group properties as goals, speed, and density as if it were individual behaviors. Moreover, the subdivision can be followed by group merging, and it happens only depending on the space restrictions. Unlike the approach proposed by Xiong et al. [14] of defining specific regions where the multi-level behavior can be achieved, in our model, the split and merge behaviors are dynamically adapted to the complexity of the environment. We also seek to represent our legions organically, compared to the usual methods to visualize macroscopic crowds [12], [7], as discussed in Section IV.

A. Multi-level control

In real life, large crowds can be divided into smaller groups to get around obstacles. Indeed, smaller groups accommodate better in free space when competing for regions with other groups. After a crowd overcomes an obstacle, groups can merge back into a single large crowd, if there is free space.

Our legion model uses BioCrowds [3], as BioClouds [12], which means that we also use space markers to provide motion to our legions (as briefly explained in Section II).

B. Splitting

Legion splitting occurs whenever a legion is in a situation (time t) where, in a future moment (time $t+1$), it might have less space to accommodate the Legion than currently desired. For example, when a group moves towards an obstacle, that obstacle takes up space, the group cannot occupy, thereby reducing its future space availability.

Each legion i has following attributes: the number of agents it represents (A_i); the desired density (D_i in people$/m^2$); speed (S_i in m/s); legion goal $\vec{G}_i$; and a perception radius (R_i) equal to the corresponding radius to guarantee the desired density D_i. In addition, as in BioClouds [12], the entities evolve in a discretized space represented as a regular grid, which cells, $M_{i,t}$, are possessed by the legion i at each frame t. As long as the group has the same amount of agents, the same density and space desired to accommodate the structure, R_i, will not be changed. In a situation of division, that is, the space needed in the $t+1$ is less than the desired for the crowd to move. This structure is subdivided into 7 new structures in the location of its current area, 6 radial legions and a central one, in a hexagonal pattern. Six new 7 legions are located at the edge of the original and are more likely to capture space than the single cloud. The radius of the newly created clouds is one-third of the original radius of the clouds. Figure 1 shows a diagram of the cloud's split pattern. The radial arrangement was chosen to facilitate the creation of smaller clouds in a pattern that best mimics the original density and space occupation. Although we chose a seven-way split for this work, the number of clouds created during a split event can be adapted to better simulate each possible experiment.

The decision to split a legion i at time $t+1$, with A_i agents, position $\vec{X}_{i,t}$, set of captured cells $M_{i,t}$, movement vector $\vec{V}_{i,t}$ and perception radius R_i is based on the available space for the legion i at frame $t+1$. We check for future available space by projecting the future position of the legion $\vec{X}_{i,t+1}$:

$$\vec{X}_{i,t+1} = \vec{X}_{i,t} + \vec{V}_{i,t}. \quad (1)$$

Then we check if the markers[1] within radius R_i around $\vec{X}_{i,t+1}$ are in the set $M_{i,t}$ or free (i.e., do not attributed to any other legion or obstacle). If this condition is true, no split is necessary. To avoid computing the split decision, at each frame, we do it at every f frames. We have found $f = 15$ produces good visual results.

If a split happens, each one of the newly created legions (in present case 7 legions) receives a new ID and inherits the same velocity, goal, and desired density as the parent legion. Moreover, R is calculated based on the number of agents assigned to each Legion ($1/7$ of parent legion) and the desired density. The set of captured cells M_t and the motion vector $\vec{V}_t$ are calculated to provide the movement of that Legion at frame $t+1$ guiding to goal $\vec{G}$. Since the cloud division can occur recursively, each structure can be divided again until the resulting structure represents only one individual.

C. Merging

Cloud merging behavior aims to regroup pair of legions (i and j) when they have the same goals ($\vec{G}_i$ and $\vec{G}_j$) and are close enough to be reunited. We determine the second rule through a simple heuristic: we consider the markers that are close to a straight line from the two clouds centers ($\vec{X}_{i,t}$ and $\vec{X}_{j,t}$), if all such markers are part of subsets M_i or M_j then two clouds are adjacent and can be reunited. If this condition is not true, there is space between the two clouds, not indicating that they are adjacent, so merge behavior is not applied. Legion merging conditions are tested at each simulation step.

[1]Markers are dots on the floor in a simulated environment. They are used in BioCrowds [3], BioClouds [12] and in the current method to discretize the space, but continuous simulation are also possible

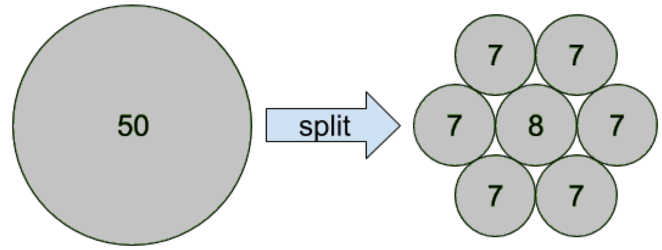


Fig. 1. A split event: a 50-agent legion splits into 7 smaller legions. The new central legion receives the remaining agents in case of fractional divisions.

A legion that is in a situation that leads to splitting and also merging would cause a not desirable repetition of splitting and merging behaviors. To avoid such sequential operations, we add a settling time St_i for each newly created legion, either by splitting or merging operations. This structure is defined as: $St_i = \{st_i, mt_i\}$, where st_i and mt_i state for last split and last merged frame, respectively. In addition, we define two other thresholds: $shift_{split}$ and $shift_{merge}$, which state for a number of frames where a new split/merge cannot happen. Such parameters work in a way that only in simulation frame $f = st_i + shift_{merge}$, a new merge can occur for legion i. Similar process happens to activate again the split operation, i.e., when simulation $f = mt_i + shift_{split}$. In this work we empirically chosen $shift_{merge} = shift_{split} = 30$ simulation frames because it results in visually coherent simulations. This value is equivalent to 3 seconds of simulation time, and can be changed to better fit different environments. When two legions merge, their data is combined to define a new legion with a new ID.

D. The control algorithm

At the beginning of the simulation, legions are generated based on user input, who also defines the environment and the obstacles. For each legion i at $t = 1$, the user defines:

1) number of agents A_i,
2) initial position $\vec{X}_{i,t}$,
3) desired density D_i (people$/m^2$),
4) desired speed S_i (m/s), and
5) goal $\vec{G}_i$ (X,Y,Z) location.

Based on these data, our method computes the perception radius R_i that the legion must have to achieve density D_i, selects the set $M_{i,t}$ with the markers inside R_i that could be attributed to legion i, and finally computes the movement vector $\vec{V}_{i,t}$ based on S_i, $\vec{G}_i$ and $M_{i,t}$.

Each legion is subdivided in 7 other legions when the split behavior occurs. Let us consider legion k as one of the 7 new legions generated from i. The data used to instantiate k is:

1) number of agents (int) $A_k = A_i/7$,

2) initial position
$\vec{X}_{k,t} = \vec{X}_{i,t} + [(\sin(2\pi k/6), \cos(2\pi k/6)) \times R_i/2]$,
3) desired density $D_K = D_i$,
4) desired velocity $S_k = S_i$, and
5) goal $\vec{G}_k = \vec{G}_i$.

As for legion i, the method computes the perception radius R_k, $M_{k,t}$, and finally $\vec{V}_{k,t}$. The new structures cannot merge during 30 frames, i.e., 3 seconds of simulation time, as defined in Section III-C. During the simulation, legions evolve in the environment, avoiding collision with other legions and obstacles (when space to move is not free). Therefore, they can split again and successively, until the Legion has only one individual.

For merge situations, as described before, two legions i and j can merge if they have same goals and be adjacent. In this case, their data is gathered to compose a new legion (e.g., m), where data is instantiated as follows:

1) number of agents $A_m = A_i + A_j$,
2) initial position $\vec{X}_{m,t} = \frac{\vec{X}_{i,t}+\vec{X}_{j,t}}{2}$,
3) desired density $D_m = D_i$,
4) desired velocity $S_m = S_i$, and
5) goal $\vec{G}_m = \vec{G}_i$.

As in the split situation, the method computes R_m, $M_{m,t}$ and finally $\vec{V}_{m,t}$. Again, legion m cannot split during 30 frames as defined in Section III-C. During such time, the legion m can keep merging again and successively, until there is not other legion that satisfies the conditions to merge.

IV. Blobby Visualization of Legions

One of the main criticisms of macroscopic crowd simulation models is the obfuscation of the crowd visualization, in the sense that the models do not inherently provide visualization for finer-grained agents [9], [10]. In this section, we present a new approach to improve the visualization of huge crowds towards a more organic appearance. Our approach is to represent both legions and the environment implicitly as density fields using a global blobby model [11].

A. Implicit regions

Implicit surfaces [15], [16] are a standard formulation for representing shapes of density field data. An *implicit region* in the plane is the set of points satisfying an inequality:

$$R = \{p \in \mathbf{R}^2 : F(p) \geq 0\},$$

where $F : \mathbf{R}^2 \to \mathbf{R}$ is a *scalar field* on the plane. When F is a regular function, the boundary of the region R is the implicit curve given by $F(p) = 0$. The inequality is $F(p) \geq 0$ quite general. Other, equivalent definitions use variants such as $F(p) \geq T$ or $F(p) \leq T$, where T is a user-controlled parameter.

Implicit regions (and solids in space) are natural primitives for constructive solid geometry [17]. Organically-looking shapes can be defined implicitly by combining simple geometric primitives [18], [19].

A fairly intuitive way to create shapes is to start with simple *skeletal shapes*, such as points, line segments, disks, and rectangles. Given a skeleton S, we let $F(p) = g(d(p,S))$, where $d(p,S)$ is the distance of p to S and g is a *potential function* used to modulate the distance [16, chapter 9]. For a point skeleton $S = \{p_0\}$ and $g(d) = d^2 - r^2$, we get the family of disks centered at p_0, as r varies. Several potential functions have been proposed. Blinn [11], in a seminal paper, used an exponential potential for his *blobby* molecules, $g(d) = \exp(-ad^2)$. Nishimura et al. [20] used piecewise quadratics for their *metaballs*. Wyvill et al. [21] used piecewise cubics for their *soft objects*. General primitives and combination methods were discussed by Bloomenthal et al. [18], [19].

B. Our implicit model

We describe the shape of the legions in an environment implicitly as follows:

$$F(x,y) = \sum_{i=1}^{n} C_i(x,y) - E(x,y) \geq T, \tag{2}$$

where C_i describes the shape of legion i, E describes the shape of the environment, and T is a threshold that allows the user to control the final shape. We shall discuss below our chosen expressions for C_i and E.

Note that the value for the environment is subtracted from the combined values for the legions. This algebraic device reflects the intuition that the obstacles in the environment exert 'pressure' on crowds to avoid that region of space, even before a crowd collides with an obstacle.

C. Modeling legions

We define the shape of legion i implicitly using an inverse square potential, like those in gravitation and electrostatics:

$$C_i(x,y) = \frac{r_i^2}{(x-x_i)^2 + (y-y_i)^2}, \tag{3}$$

where r_i is the current radius of legion i and (x_i, y_i) is its center. When there is only one legion and no obstacles, we have $n = 1$ and $E(x,y) = 0$, and so $F(x,y) \geq T$ describes a disk of radius r_1 when $T = 1$. When there several legions and no obstacles, $F(x,y) \geq T$ describes shapes that merge the disks organically as T varies (Figure 2).

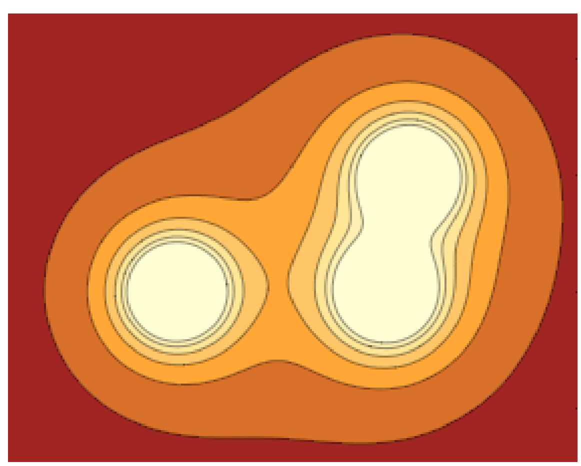


Fig. 2. Isolines representing a blobby model for several threshold values.

D. Modeling the environment

We support two implicit models for the environment. In the first model, the obstacles have a simple geometry, like a rectangular or cylindrical column in a room, and we can describe them analytically. More precisely, we define the shape of obstacle j implicitly as follows:

$$O_j(x,y) = e^{-D_j(x,y)}, \quad (4)$$

where $D_j(x,y)$ is the distance of the point (x,y) to obstacle j.

The shape function for the environment is then:

$$E(x,y) = \sum_{j=1}^{m} O_j(x,y) \quad (5)$$

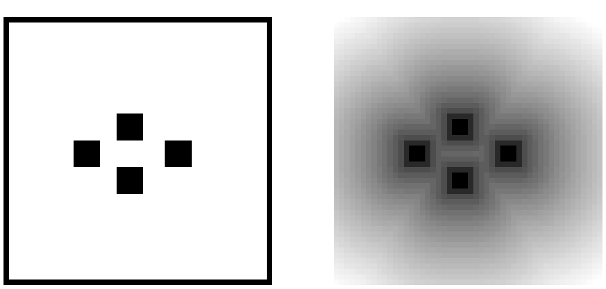
The obstacles can have any geometry, as long as we can compute their distance function [22, chapter 6]. Note that the functions for obstacles decay faster than the functions for legions. This allows legions to connect from a more significant distance, before obstacles can block legions from entering a particular space.

In the second model for the environment, we describe its walkable and non-walkable areas as a global distance function. We map the distance of each point in space to the closest obstacle using a discretized distance transform [23] applied to a binary image describing the environment: white pixels for walkable markers and black pixels for obstacle occupied markers. The resulting distance values are capped at a user-selected maximum value, representing the distance at which the influence of the obstacle on crowds is considered negligible, and normalized into the $[0,1]$ range. The normalized values are encoded into a gray-scale image that describes the obstacle-distance at each discretized marker.

The shape function for the environment is then:

$$E(x,y) = \frac{1}{s \cdot D(x,y)}, \quad (6)$$

where $D(x,y)$ is obtained by sampling the image-encoded distance map at the pixel corresponding to the marker closest to the point (x,y), and s is the scale parameter value to denormalize the sampled distance map (s can also be used to scale the strength of the environmental pressure on agents). Figure 3 illustrates an environment image and its distance map.



(a) (b)

Fig. 3. (a) The environment image of experiment B. Walkable terrain in white; obstacles in black. (b) The corresponding distance map. Lighter colors mean greater distance to the closest obstacle.

V. Experimental Results

We now present some experimental results obtained with our model. The experiments focus on exploring the new multi-level behavior and the proposed visualization. Experiments A and B compare how different complexities of environment (varied amount of obstacles) affect split and merge behaviors. In order to compare two different scenarios, the number of split and merge operations were counted during each experiment, as well as the crowd densities and average velocities. Experiments A and B have the following data:

- The simulation runs at 10 frames per simulated second.
- The cells in the regular grid are $0.125m$ size and area of $0.015625m^2$;
- Each experiment has, at the beginning, one only legion containing 100 agents with preferred density of $1\text{agent}/m^2$;
- desired speed $S = 1.3m/s$;
- goal position $\vec{G} = (5,25)$;
- obstacle size $5mx5m$, where experiment A has one obstacle positioned at $(15,25)$, and experiment B has four obstacles positioned at $(15,25)$, $(23,30)$, $(23,20)$, $(32,25)$.
- total size of environment $100mx50m$.

A. Emergence of split and merge behaviors

Firstly, we want to show the impact of environmental complexity on the emergence of split and merged behaviors. To the best of our knowledge, there is no dynamically hybrid crowd simulation methods which to compare to, so we compare results to the purely macroscopic simulation. In experiment A, 1 split and 6 merge operations were executed; while in experiment B, 11 split operations and 50 merge operations happened. Table I compares the two experiments. Merge operations occur more often than split operations since one split operation instantiates 7 legions, which need six merge operations to merge back into one Legion. We emphasize that split/merge operations are asymmetrical, i.e., one Legion can split one time and gives origin to 7 legions, that have to merge in pairs from 4 to 6 times to become the same Legion again.

TABLE I

RESULTS OF EXPERIMENTS A AND B: AVERAGE DENSITY (AGENTS$/m^2$), AVERAGE SPEED (m/s), NUMBER OF SPLITS AND MERGES.

	DENSITY	SPEED	# SPLITS	# MERGES
A	1.001	1.30	1	6
B	1.003	1.29	11	50

Figures 4 and 6 show snapshots of experiments A and B: a legion recursively splitting and merging to traverse a scenario with obstacles. In experiment B, the first Legion never fully regrouped into a single legion again, since the created small groups got far away from others due to obstacles. This fact is consistent with real-life: people can drift apart due to space restrictions, and lose group identity. Figure 7 shows the same scenario simulated with microscopic BioCrowds behaviors.

It is essential to mention that we decide not to include path planning for all the evaluated simulations, so the only input was the goal and not a path to achieve the goal. We did that because we do not want to interfere with the way the legions will avoid the obstacles, deciding, for instance, going through the right or left side of the obstacles. In order to compare with BioCrowds, we also turned off the path planning, and as a consequence, we can see some agents in a local minimum, behind the obstacles. We compared our results with BioClouds [12], in the environment of experiment B, as can be seen in Figure 5. We measured the average density (people$/m^2$) along the simulations for the three methods: Legion ($\bar{d} = 1.03$), BioCrowds ($\bar{d} = 1.23$), and BioClouds ($\bar{d} = 0,99$). Even though the results are similar, as expected, BioCrowds achieved higher average density, because at the end agents were very close to each other to achieve the same goal. On the other hand, BioClouds kept lower densities because it is always one circle trying to achieve the desired density. Legion achieved an average density close to the desired one, even after splits and merges.

B. Blobby visualization of legions

We now illustrate our improved blobby visualization of crowds. Experiments A and B are repeated with the improved visualization, and the resulting simulations are presented. The results are evaluated qualitatively.

We propose a density heat map visualization (Figure 6) and a visualization including BioCrowds. While the heat map aimed to represent current data relative to the described crowd, i.e., how compact people are behaving in a given scenario, it could not provide much data on the interactions among the agents contained in a crowd. The shape of a legion is approximated by a circle for space capturing behavior, which leads to legions that look a lot more rounded than a real-life crowd would naturally organize itself to be (Figure 4). The split and merge behaviors of legions increase the number of crowd interactions in a simulation compared to the highest level of abstraction of the BioClouds model. However, when new groups are generated, their shapes are still rigidly approximated by circles, failing to model how people spread to occupy the environment during interactions, as in Figure 6.

Our blobby visualization of legions and obstacles aims at a more organic-looking cloud environment occupation. This is illustrated in Figures 9 and 10 for experiment A and Figure 8 for experiment B. Figure 9 is rendered using the analytical description of obstacles (eq. 5) and Figure 10 is rendered using the distance map description of the environment (eq. 6). Figures 6 and 8 show a comparison of our proposed blobby visualization and the original BioClouds heatmap visualization.

VI. Final Remarks

In this paper, we proposed solutions for two common issues in the macroscopic crowd: the multi-level control and a new organic visualization. The multi-level control was implemented as a function of split/merge behaviors by increasing the number of inter-entity interactions. This approach should better mimic crowd interactions present in microscopic simulations, leveraging the simulation's complexity by varying the granularity of the simulation dynamically. The proposed blobby visualization aimed to propose a more organic visualization of huge crowds. We modeled legions and obstacles in the environment as implicit regions combined as a global blobby object. The resulting visualization provides a more organic appearance to how crowds fit the environment. One open question is how to ensure that the different shapes of space occupation provided by blobs are still representative of the simulated crowd densities. This is currently the object of our investigation.

Acknowledgements

The authors are partially supported by CAPES, FAPERGS and CNPq research grants.

References

[1] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, p. 4282, 1995.
[2] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*. Springer, 2011, pp. 3–19.
[3] A. de Lima Bicho, R. A. Rodrigues, S. R. Musse, C. R. Jung, M. Paravisi, and L. P. Magalhães, "Simulating crowds based on a space colonization algorithm," *Computers & Graphics*, vol. 36, no. 2, pp. 70–79, 2012.
[4] S. Patil, J. van den Berg, S. Curtis, M. Lin, and D. Manocha, "Directing crowd simulations using navigation fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 2, pp. 244–254, 2011.
[5] D. Helbing, "A fluid dynamic model for the movement of pedestrians," *Complex Systems*, vol. 6, no. 5, pp. 391–415, 1992.
[6] A. Treuille, S. Cooper, and Z. Popović, "Continuum crowds," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 1160–1168, 2006.
[7] R. Hughes, "The flow of human crowds," *Annu. Rev. Fluid Mech.*, vol. 35, no. 1, pp. 169–182, 2003.
[8] V. J. Cassol, E. Smania Testa, C. Rosito Jung, M. Usman, P. Faloutsos, G. Berseth, M. Kapadia, N. I. Badler, and S. Raupp Musse, "Evaluating and optimizing evacuation plans for crowd egress," *IEEE Computer Graphics and Applications*, vol. 37, no. 4, pp. 60–71, 2017.
[9] D. Thalmann and S. R. Musse, *Crowd Simulation*. Springer, 2012.
[10] N. Pelechano, K. O'Brien, B. Silverman, and N. Badler, "Crowd simulation incorporating agent psychological models, roles and communication," Center for Human Modeling and Simulation, Tech. Rep., 2005.
[11] J. F. Blinn, "A generalization of algebraic surface drawing," *ACM Transactions on Graphics*, vol. 1, no. 3, pp. 235–256, 1982.
[12] A. D. S. Antonitsch, D. H. M. Schaffer, G. W. Rockenbach, P. Knob, and S. R. Musse, "Bioclouds: A multi-level model to simulate and visualize large crowds," in *Advances in Computer Graphics. CGI 2019. Lecture Notes in Computer Science, vol 11542*. Springer, 2019, pp. 15–27.
[13] S. Narang, A. Best, S. Curtis, and D. Manocha, "Generating pedestrian trajectories consistent with the fundamental diagram based on physiological and psychological factors," *PLOS ONE*, vol. 10, no. 4, 2015.
[14] M. Xiong, M. Lees, W. Cai, S. Zhou, and M. Y. H. Low, "Hybrid modelling of crowd simulation," *Procedia Computer Science*, vol. 1, no. 1, pp. 57–65, 2010.
[15] J. Bloomenthal, Ed., *Introduction to Implicit Surfaces*. Morgan Kaufmann, 1997.
[16] A. Gomes, I. Voiculescu, J. Jorge, B. Wyvill, and C. Galbraith, *Implicit Curves and Surfaces*. Springer, 2009.
[17] A. Ricci, "A constructive geometry for computer graphics," *The Computer Journal*, vol. 16, no. 2, pp. 157–160, 1973.
[18] J. Bloomenthal and B. Wyvill, "Interactive techniques for implicit modeling," *ACM SIGGRAPH Computer Graphics*, vol. 24, no. 2, pp. 109–116, 1990.
[19] J. Bloomenthal and K. Shoemake, "Convolution surfaces," *ACM SIGGRAPH Computer Graphics*, vol. 25, no. 4, pp. 251–256, Jul. 1991.
[20] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakawa, and K. Omura, "Object modeling by distribution function and a method of image generation." *Trans. IECE Japan*, vol. J68-D, pp. 718–725, 1985.
[21] G. Wyvill, C. McPheeters, and B. Wyvill, "Data structure for soft objects," *The Visual Computer*, vol. 2, no. 4, pp. 227–234, 1986.
[22] P. J. Schneider and D. H. Eberly, *Geometric Tools for Computer Graphics*. Elsevier, 2003.
[23] S. Gustavson and R. Strand, "Anti-aliased Euclidean distance transform," *Pattern Recognition Letters*, vol. 32, no. 2, pp. 252–257, 2011.

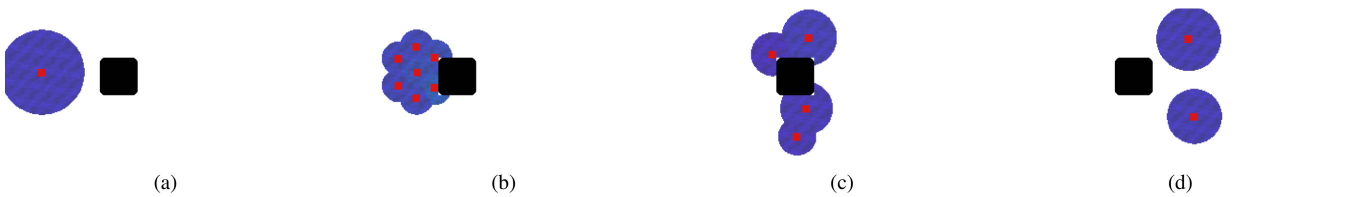


Fig. 4. Snapshots from experiment *A* using the heatmap visualization. (a) A legion heads towards an obstacle. (b) The legion splits due to the obstacle. (c) The legions go around the obstacle. (d) The legions begin regrouping.

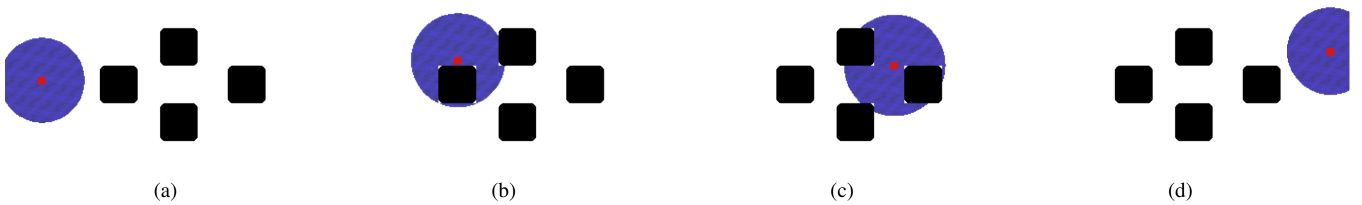


Fig. 5. Snapshots from experiment *B*, simulated with BioClouds. (a) A cloud heads towards an obstacle. (b) The cloud avoids the obstacle and increases its perception radius. (c) The cloud avoids another obstacle. (d) The cloud moves towards the goal and returns to its original size.

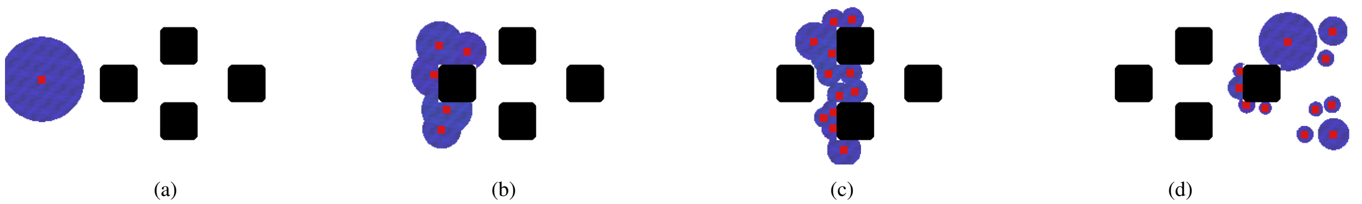


Fig. 6. Snapshots from experiment *B* using the heatmap visualization. (a) A legion heads towards the obstacles. (b) The legion splits due to the obstacles. (c) The legions go around the obstacles. (d) The legions begin regrouping.

Fig. 7. Snapshots of experiment *B* simulated with BioCrowds. A crowd of BioCrowds agents separating and regrouping when interacting with the obstacles. (a) A crowd splits when interacting with an obstacle. (b) The crowd goes around the obstacle. (c) The crowds start grouping back together. (d) The crowds continue grouping back together.

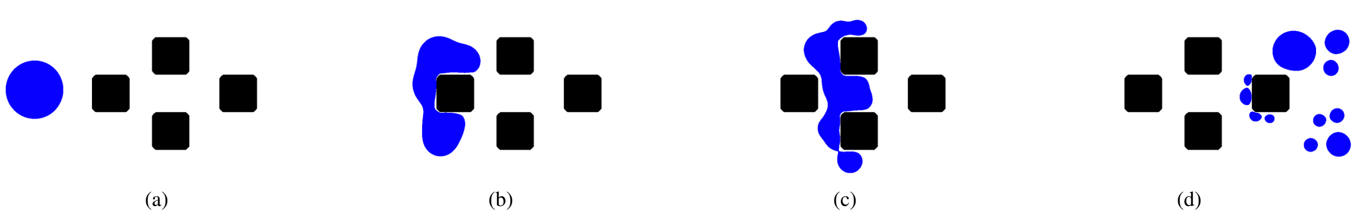


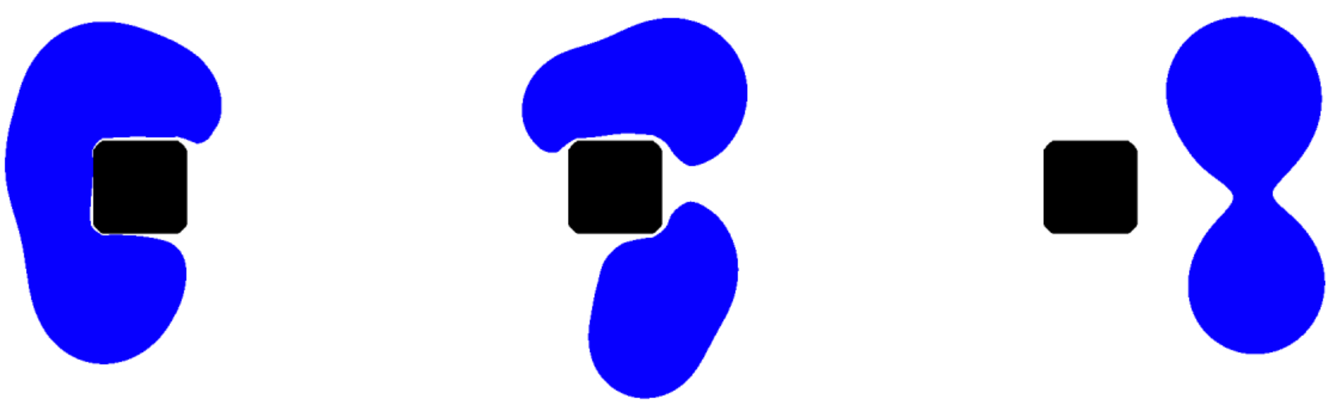
Fig. 8. Snapshots from experiment *B* using our blobby visualization. (a) A legion heads towards the obstacles. (b) The legion splits due to the obstacles. (c) The legions go around the obstacles. (d) The legions begin regrouping.

(a) A legion splits when interacting with an obstacle.

(b) The legions go around the obstacle.

(c) The legions start merging back together.

Fig. 9. Snapshots of experiment A. Legions split and merge when interacting with an obstacle. Legions are rendered using eq. 5. The blue regions show the area occupied by the crowd.



(a) A legion splits when interacting with an obstacle.

(b) The legions go around the obstacle.

(c) The legions start merging back together.

Fig. 10. Snapshots of experiment A. Legions split and merge when interacting with an obstacle. Legions are rendered using eq. 6. The blue regions show the area occupied by the crowd.