Priority driven Local Optimization for Crowd Simulation

Extended Abstract

Himangshu Saikia
KTH Royal Institute of Technology
Stockholm, Sweden
saikia@kth.se

Fangkai Yang
KTH Royal Institute of Technology
Stockholm, Sweden
fangkai@kth.se

Christopher Peters
KTH Royal Institute of Technology
Stockholm, Sweden
chpeters@kth.se

ABSTRACT

We provide an initial model and preliminary findings of a lookahead based local optimization scheme for collision resolution between agents in large goal-directed crowd simulations. Considering crowd simulation to be a global optimization problem, we break down this large problem into smaller problems where each potential collision resolution step is independently optimized in terms of a *criticality* measure. Agents resolved earlier in order of criticality, maintain the optimized velocity obtained, for the resolution of agents that come later in that order. Hence, the problem is converted to a low dimensional optimization problem of one or two agents where all other obstacles are static or deterministically dynamic. We illustrate the performance of our method on four well known test scenarios.

KEYWORDS

agent-based crowd navigation; collision avoidance; optimization

ACM Reference Format:
Himangshu Saikia, Fangkai Yang, and Christopher Peters. 2019. Priority driven Local Optimization for Crowd Simulation. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION AND RELATED WORK

Two main categories in Crowd Simulation techniques exist - the Macroscopic and the Microscopic [12]. Several approaches have been proposed to simulate crowds in both the macroscopic and microscopic categories [17]. The macroscopic methods [3, 4, 8, 13] simulate crowds as continuum dynamics, often in large crowded cases, ignoring individual behaviors and interactions. The microscopic methods or agent-based methods [2, 15, 19], focus on behaviors and interactions of each individual from a local agent-centric standpoint.

Two of the earliest agent-based models due to Boids [11] and Helbing et al. [2] were based on physical forces between agents. These methods however, suffered from a lack of anticipatory actions ahead of likely collisions. Anticipatory algorithms using a time to collision (ttc) strategy [5, 6] and ones involving a minimal predicted distance [9, 10] have been seen thereafter. Other velocity based algorithms [14, 15] select collision-free velocities based on a cost function on the velocity space of all agents. Recently, learning based approaches have also gained popularity with human trajectory prediction being done using Long Short Term Memory networks [1, 16] and improvements to them using perception models [18] have also been proposed.

Our method is an agent-based goal-directed method where the objective is for all agents to reach their desired targets while at the same time avoiding collisions with static (walls) or dynamic (other agents) obstacles. This problem can be viewed as a global optimization problem, where the entire configuration of agents supposedly affects the individual decisions for each agent. However, from observations in the real world we know that this is not the case. In most cases, small groups of agents collaboratively resolve close to imminent collisions between them. Our method adopts such a collaborative effort between small groups of agents and transforms the global problem into a local one. We introduce the notion of *criticality* - the notion that defines whether two agents would come dangerously close to each other. We use a Particle Swarm Optimization[7] strategy to resolve one or two critical agents at a time and keep resolving these small local batches until all agents are resolved.

2 METHOD

2.1 Criticality

Two agents A_1 and A_2 are said to be critical if their positions and velocities maintain the following relations

$$\hat{\ell}_{ij} = |\mathbf{x}_1 - \mathbf{x}_2| \sqrt{1 - \left(\frac{(\mathbf{x}_1 - \mathbf{x}_2) \cdot (\mathbf{v}_1 - \mathbf{v}_2)}{|\mathbf{x}_1 - \mathbf{x}_2||\mathbf{v}_1 - \mathbf{v}_2|} \right)^2} \leq d_c \qquad (1)$$

$$\hat{t}_{ij} = -\frac{(\mathbf{x}_1 - \mathbf{x}_2) \cdot (\mathbf{v}_1 - \mathbf{v}_2)}{|\mathbf{v}_1 - \mathbf{v}_2|^2} \geq 0 \qquad (2)$$

where $\mathbf{x}_i$ and $\mathbf{v}_i$ denote the position and velocity of agent A_i and d_c denotes the critical distance - the comfortable distance that agents like to maintain within them. In our experiments we set $d_c = 3r_A$ where r_A is the agent radius or agent size. Equation 1 states that the minimum approachable distance $\hat{\ell}_{ij}$ between two agents must never be greater than the critical distance and is hence referred to as the *critical distance condition.* Equation 2 states that the time to reach the minimum approachable distance $\hat{t}_{ij}$ must be non negative (i.e., must happen in the future), and hence is referred to as the *critical time condition.* These conditions are derived considering agents as circular disc shaped objects. Similar conditions are derived for agent-wall collisions as well by treating walls as line-segments.

2.2 Criticality Resolution

We use Particle Swarm Optimization (PSO) to resolve critical agents by altering their velocities by applying a combination of a rotational transformation (turn) and a reverse acceleration (brake). To denote

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada. © 2019 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

these two values we use $a_{\downarrow}$ for the braking strength and $a_{\leftrightarrow}$ for the rotating strength. We call $(a_{\downarrow}, a_{\leftrightarrow})$ as an alteration strength pair. With $\theta = \pi a_{\downarrow}/2$ and $\phi = \pi a_{\leftrightarrow}/2$, the velocities are modified as

$$\mathbf{v}_{i,t} = (1 + \sin\theta)\begin{bmatrix}\cos\phi & -\sin\phi \\ \sin\phi & \cos\phi\end{bmatrix}\mathbf{v}_{i,t-1} \tag{3}$$

When agents are not critical, they automatically 'relax' to their desired speed s_d aligned to their respective targets $\mathbf{T}_i$ as

$$\mathbf{v}_{i,t} = \lambda\mathbf{v}_{i,t-1} + (1-\lambda)s_d\frac{(\mathbf{T}_i - \mathbf{x}_{i,t})}{||\mathbf{T}_i - \mathbf{x}_{i,t}||} \tag{4}$$

where $\lambda \in [0, 1]$ is a velocity relaxation parameter.

2.3 Configuration Score

A particular configuration of agents is scored higher if fewer agents are critical and vice-versa. It is computed using a lookahead parameter τ which simulates the crowd τ time-steps into the future and relaxing velocities for every time-step using Equation 4. The configuration score for a pair of agents A_i and A_j is given as

$$s_{c,ij} = \begin{cases}1, & \text{if } \hat{t} < 0 \vee \hat{\ell}_{ij} \geq d_{crit} \\ -\frac{k}{(\hat{\ell}_{ij}+\epsilon)(\hat{t}+1)}, & \text{otherwise}\end{cases}$$

Here the constants are chosen to be $k = 100$ and $\epsilon = 10^{-4}$. A higher value of k is to ensure that critical agents are heavily penalized. A small value of epsilon is necessary to prevent division by zero cases.

The net configuration score is the sum of scores for all agent-agent and agent-wall configuration pairs as well as an energy term which penalizes high values of $a_{\leftrightarrow}$ and $a_{\downarrow}$.

2.4 Algorithm

Our algorithm works on an initial configuration of agents and walls and iterates for as many time-steps as required for all agents to reach their targets. A computation of a single time-step can be formulated as follows

(1) Define a configuration as a set of agents and walls as $C = (\mathcal{A}, \mathcal{W})$.
(2) Initialize a list of processed agents $\mathcal{P} = \emptyset$, a list of to be resolved agents $\mathcal{R} = \emptyset$ and a best particle vector $\mathcal{H}_{best}$ initialized with zero alteration strength pairs.
(3) Find the agent-agent pair (A_i, A_j) where $A_i \notin \mathcal{P}$ which is the most critical in the current configuration C i.e., has the minimum $\hat{t}$. If $A_j \notin \mathcal{P}$, update $\mathcal{R} \leftarrow \mathcal{R} \cup \{A_i, A_j\}$, otherwise update $\mathcal{R} \leftarrow \mathcal{R} \cup \{A_i\}$. It is also possible that an agent-wall pair (A_i, W) has the lowest $\hat{t}$. In that case, update $\mathcal{R} \leftarrow \mathcal{R} \cup \{A_i\}$. If no critical agents are found, go to Step (6) and perform a move.
(4) Run the PSO algorithm on the subset $\mathcal{R}$ of critical agents that need to be resolved. The particle vector dimension is $2|\mathcal{R}|$, for two alteration pairs per critical agent $A_i \in \mathcal{R}$. Since it is guaranteed that $|\mathcal{R}| \leq 2$ the optimization remains low-dimensional. We obtain $\mathcal{H}_{curr}$ after convergence.
(5) Assign the corresponding entries for $\mathcal{H}_{best,i} := \mathcal{H}_{curr,i} \forall A_i \in \mathcal{R}$. Update $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{R}$ and set $\mathcal{R} = \emptyset$. Repeat Step (3) until all agents are processed.
(6) Make a move by altering the current velocities using the alteration strengths in $\mathcal{H}_{best}$. Repeat Step (1) for the next time-step, and continue until all agents reach their targets.

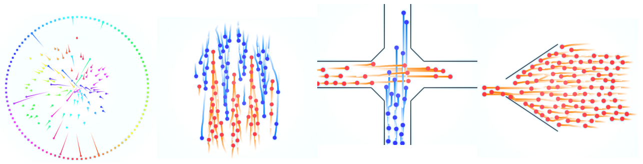
(a) Circle (b) Hallway (c) Orthogonal Corridor (d) Bottleneck

Figure 1: Illustration of the four test scenarios and a snapshot of the simulation at a random time-step.

3 RESULTS AND EVALUATIONS

We tested our method on four test scenarios that exemplify several important features necessary for a successful crowd navigation. We consider a look-ahead value of $\tau = 10$ which we found to be optimal for all of our scenarios. We choose $r_A = 0.25m$ and $s_d = 0.2m/s$ in all our experiments.

The results are shown in Figure 1. In the *circle* scenario, agents are placed equidistant from each other on the periphery of a circle with their corresponding targets located on diametrically opposite points. Agents in our method exhibit conscientious behaviour, in that they anticipate impending collisions way before they occur. In the *hallway* scenario, two large groups of agents approach each other from opposite ends in a large hallway. The formation of lanes is a prerequisite for efficient collision free navigation in this scenario and they are clearly seen in our method. The *orthogonal corridor* scenario involves two sets of agents walking on two orthogonal hallways from one side to the other. The emergence of 45° lanes is a known emergent behaviour in this scenario necessary for efficient navigation. Such lanes are clearly seen in our simulation as well. The *bottleneck* scenario requires a set of agents to pass through a narrow bottleneck on the way to their targets. Here as well, agents in our method exhibit anticipatory behaviour towards future congestion and modify their velocities to enable all agents to comfortably maneuver through the bottleneck.

4 DISCUSSION AND LIMITATIONS

We introduced the concept of *criticality* and using this concept we presented an efficient crowd navigation strategy. However, our method is not real-time beyond ~ 100 agents and the many small optimizations that it has to perform at each time-step is the key reason for this slow performance. We hope to find a balance between the dimensionality of our local optimization (number of agents to resolve together) and the number of such optimizations in total. We think this balance is key to scaling our method to deal with even larger crowds.

We also hope to evaluate our method with state-of-the-art methods that have been proposed in the recent past and report quantitative scores on important features of a crowd simulation such as anticipation of congestion, orientation to targets and distances maintained with other agents and obstacles.

REFERENCES

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. 2016. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 961–971.

[2] Dirk Helbing and Peter Molnar. 1995. Social force model for pedestrian dynamics. *Physical review E* 51, 5 (1995), 4282.

[3] Roger L Hughes. 2002. A continuum theory for the flow of pedestrians. *Transportation Research Part B: Methodological* 36, 6 (2002), 507–535.

[4] Roger L Hughes. 2003. The flow of human crowds. *Annual review of fluid mechanics* 35, 1 (2003), 169–182.

[5] Ioannis Karamouzas, Peter Heil, Pascal Van Beek, and Mark H Overmars. 2009. A predictive collision avoidance model for pedestrian simulation. In *International Workshop on Motion in Games*. Springer, 41–52.

[6] Ioannis Karamouzas, Brian Skinner, and Stephen J Guy. 2014. Universal power law governing pedestrian interactions. *Physical review letters* 113, 23 (2014), 238701.

[7] James Kennedy. 2011. Particle swarm optimization. In *Encyclopedia of machine learning*. Springer, 760–766.

[8] Rahul Narain, Abhinav Golas, Sean Curtis, and Ming C Lin. 2009. Aggregate dynamics for dense crowd simulation. In *ACM transactions on graphics (TOG)*, Vol. 28. ACM, 122.

[9] Anne-Hélène Olivier, Antoine Marin, Armel Crétual, and Julien Pettré. 2012. Minimal predicted distance: A common metric for collision avoidance during pairwise interactions between walkers. *Gait & posture* 36, 3 (2012), 399–404.

[10] Julien Pettré, Jan Ondřej, Anne-Hélène Olivier, Armel Cretual, and Stéphane Donikian. 2009. Experiment-based modeling, simulation and validation of interactions between virtual walkers. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 189–198.

[11] Craig W Reynolds. 1987. Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH computer graphics*, Vol. 21. ACM, 25–34.

[12] Daniel Thalmann. 2007. Crowd simulation. *Wiley Encyclopedia of Computer Science and Engineering* (2007).

[13] Adrien Treuille, Seth Cooper, and Zoran Popović. 2006. Continuum crowds. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 1160–1168.

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

[15] Jur Van den Berg, Ming Lin, and Dinesh Manocha. 2008. Reciprocal velocity obstacles for real-time multi-agent navigation. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 1928–1935.

[16] Anirudh Vemula, Katharina Muelling, and Jean Oh. 2018. Social attention: Modeling attention in human crowds. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1–7.

[17] Ming-Liang Xu, Hao Jiang, Xiao-Gang Jin, and Zhigang Deng. 2014. Crowd simulation and its applications: Recent advances. *Journal of Computer Science and Technology* 29, 5 (2014), 799–811.

[18] Fangkai Yang, Himangshu Saikia, and Christopher Peters. 2018. Who are my neighbors?: A perception model for selecting neighbors of pedestrians in crowds. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*. ACM, 269–274.

[19] WJ Yu, R Chen, LY Dong, and SQ Dai. 2005. Centrifugal force model for pedestrian dynamics. *Physical Review E* 72, 2 (2005), 026112.