Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2016)
Ladislav Kavan and Chris Wojtan (Editors)

Dynamic Group Behaviors for Interactive Crowd Simulation

Liang He[1] and Jia Pan[2] and Sahil Narang[1] and Dinesh Manocha[1]

[1]The University of North Carolina at Chapel Hill
[2]City University of Hong Kong

http://gamma.cs.unc.edu/Proxemic/

Figure 1: Our method can generate dynamic group behaviors along with coherent and collision free navigation at interactive frame rates. We highlight the performance in a street-crossing scenario, in which different groups are shown with different colors. Our approach can automatically adapt to the environment and the number, shape, and size of the groups can change dynamically.

Abstract

We present a new algorithm to simulate dynamic group behaviors for interactive multi-agent crowd simulation. Our approach is general and makes no assumption about the environment, shape, or size of the groups. We use the least effort principle to perform coherent group navigation and present efficient inter-group and intra-group maintenance techniques. We extend the reciprocal collision avoidance scheme to perform agent-group and group-group collision avoidance that can generate collision-free and coherent trajectories. The additional overhead of dynamic group simulation is relatively small. We highlight its interactive performance in complex scenarios with hundreds of agents and highlight its benefits over prior methods.

1. Introduction

The problem of simulating the trajectories and behaviors of a large number of human-like agents frequently arises in computer graphics, virtual reality, and computer-aided design. This problem includes the generation of pedestrian movements in a shared space and the collaboration between the agents governed by social norms and interactions. The resulting crowd simulation algorithms are used to generate plausible simulations for games and animation, as well as to accurately predict the crowd flow and patterns in architectural models and urban environments.

One of the main challenges is modeling different behaviors corresponding to navigation, collision-avoidance, and social interactions that lead to self-organization and emergent phenomena in crowds. Prior research and observations in sociology and behavioral psychology have suggested that real-world crowds are composed of (social) groups. The group is regarded as a meso-level concept and is composed of two or more agents that share similar goals over a short or long period of time, and exhibit similar movements or behaviors. In many instances, up to 70% of observed pedestrians are walking in such groups [CJ61, GBS14]. As a result, it is important to model these group dynamics, including intra-group and inter-group interactions within a crowd.

In this paper, we address the problem of simulating the group behaviors that are similar to those observed in real-world scenarios. In crowds, small groups are dynamically formed as some of the agents move towards their goals and generate behaviors such as aggregation, dispersion, following the leader, etc. As the individual agents respond to a situation (e.g., panic or evacuation), the dynamic behaviors can result in the splitting of a large group or in-new groups being formed. Such group behaviors are frequently observed in public places, sporting events, street-crossings, cluttered areas where the pedestrians tend to avoid the obstacles, etc. Furthermore, the geometric shapes of the groups and the size of these groups may change. Some earlier observations have suggested that group sizes differ according to a Poisson distribution [Jam53].

DOI: 10.2312/sca.20161231

Prior work on modeling group behavior is mostly limited to cohesive movements or spatial group structures. The simplest algorithms cluster the agents into a fixed number of groups and the size of each group remains fixed (i.e. static grouping). These algorithms may not be able to model the changing shape or size of the group, the splitting of a large group into sub-groups, or the merging of small groups into a large group. Furthermore, in some scenarios an agent may switch from one group to the another group in close proximity. It is important to model such dynamic behavior in arbitrary environments.

Main Results: We present a novel algorithm to generate dynamic group behaviors using multi-agent crowd simulation. We use spatial clustering techniques to generate group assignments that take into account the positions and velocities of the agents. Our group-level navigation algorithm is based on the principle of least effort that tends to maintain the group relationship as each agent proceeds towards its goal position. We present efficient inter-group and intra-group level techniques to perform coherent and collision-free navigation using reciprocal collision avoidance. The shape and size of each group are automatically updated as new agents are assigned to the group or when some agents leave the group.

Our approach is used to compute the new adapted velocity, which is a transitory meso-level velocity for each agent that not only avoids collisions with other agents and obstacles, but also performs coherent group navigation. This makes it possible to handle high-density crowds as well as arbitrarily shaped groups at interactive frame rates. The overall approach has been implemented and we highlight its performance on many complex benchmarks with dynamic group behaviors. The additional overhead of group computation and maintenance is relatively small and our approach takes a few milliseconds per frame on scenarios with hundreds of agents. We demonstrate the benefits over prior methods based on agent-based or meso-scale algorithms and also compare the trajectory behaviors generated by our algorithm with real-world pedestrian trajectories. Overall, our approach offers the following benefits:

1. Our formulation is general and makes no assumption about the environment, size or shape of the groups.
2. We present an efficient algorithm for agent-group and group-group collision avoidance by extending the agent-agent reciprocal velocity obstacle formulation [vdBGLM11].
3. Our approach can generate dynamic group behaviors in terms of formation, merging, splitting, and re-assignment.
4. We observe plausible group behaviors and smooth trajectories, similar to those observed in real-world crowds.

The rest of the paper is organized as follows. We briefly survey prior work in crowd simulation and group behaviors in Section 2. We introduce the notation and give an overview of our approach in Section 3. The overall algorithm is described in Section 4 and we highlight its performance in Section 5.

2. Related Work

In this section, we give a brief overview of prior work on crowd simulation and group behaviors.

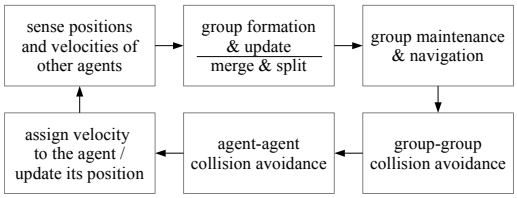


Figure 2: Algorithm pipeline: We show the various components of our algorithm for dynamic group behaviors.

2.1. Crowd Simulation

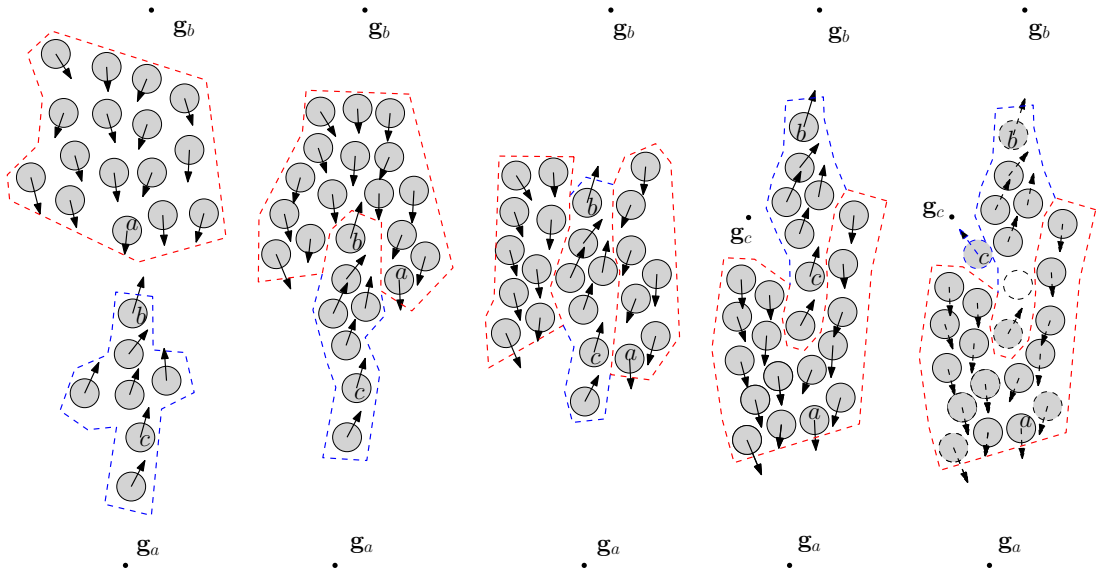
There is extensive work on modeling the behavior of crowds. This work includes multi-agent simulation techniques for computing collision-free trajectories and navigation based on social forces [HM95], rule-based methods [Rey87, PAB07], geometric optimization [vdBGLM11, YCP*08, GCK*09, NBCM15, KO12], vision-based steering [OPOD10], cognitive methods [YT07], personality models [DGABar], etc. Other classes of simulation algorithms are based on data-driven methods [LCSCO09, KCT*13] and estimations of the simulation parameters based on real-world crowd data [WGO*14, BKHF14]. The macroscopic simulation algorithms compute fields for pedestrians to follow based on continuum flows [TCP06] or fluid models [NGCL09], and are mainly used for high-density crowds.

2.2. Group Behavior Simulation

Group behaviors have been studied in computer graphics [YCP*08, LCHL07, PQC12, CSM12, HLLO10, KDB12], robotics [KLB12], pedestrian dynamics [GBS14], and social psychology [Kno73]. Prior techniques have mainly been mainly used to simulate static or fixed-sized groups, transform between two different group formations [AMBR*12, TYK*09], or perform group-based collision avoidance [SSKF10, KO04, SPN08, HKBK14, HvdB13, KO12, KG15]. It is not clear whether these methods can efficiently simulate dynamic groups of varying sizes in arbitrary environments. Golas et al. [GNCL14] have proposed a hybrid approach that combines microscopic and macroscopic methods, and generates grouping behaviors by taking into account long-range trajectory predictions. However, this approach cannot generate stable grouping behavior, and long-range prediction can be expensive. Recently, He et al. proposed a distributed following strategy [HPWMar] to handle dynamic behaviors in scenes. However, it can only handle a few agents and is unable to simulate arbitrary merging and splitting behaviors or large numbers of groups. We extend that approach to general dynamic behaviors. Godoy et al. [GKGG16] present an elegant approach to performing implicit coordination between the local motion of the agents to improve global navigation. Our clustering based approach is complementary and can be combined with this method.

3. Overview

In this section, we introduce our notation and give an overview of our approach. The various components of our approach are shown in Figure 2.



(a) group formation (b) group maintenance (c) group split (d) group re-merge (e) leave a group

Figure 3: Dynamic group behaviors: (a) The agents are clustered into groups according to their positions and velocities. Each agent has its individual goal (local or global), e.g., agent a, b, c's goals are $\mathbf{g}_a$, $\mathbf{g}_b$, $\mathbf{g}_c$ (see (e)), respectively. (b) After a while, two groups will run into each other, but both groups will maintain their behaviors during navigation. (c) For collision-free navigation, one group (marked by the red dashed line) is split into two groups (splitting). (d) After these two groups pass through each other, the split groups merge back into a single group (merging). (e) If one agent in a group can approach its goal easily, it will choose to leave the group and navigate alone.

3.1. Dynamic Grouping Behaviors

Our approach is designed for multi-agent crowd simulation algorithms. We assume that during each step of the simulation, each agent in the crowd has an intermediate goal position that is used to compute its adapted velocity. This goal position can change over the course of the simulation. The notion of dynamic grouping is motivated by real-world crowd observations. Many studies have highlighted the importance of group dynamics in the context of modeling the interactions between some agents and other agents, and between agents and objects in the environment [Rei01]. The number of such groups or the size of each group (i.e. number of agents) can change during the course of the simulation.

The dynamic grouping behavior within a crowd is classified based on how the agents are dynamically clustered into groups. Given a set of n independent agents sharing a (2D) environment consisting of obstacles, we automatically compute these groups using spatial and temporal clustering algorithms. In particular, given the current position $\mathbf{p}_a$ and velocity $\mathbf{v}_a$ of an agent a, we need to cluster all agents into a set of groups $\{G^i\}$ according to a pairwise similarity metric defined over the agents. It is possible that some agent may not belong to any group and is treated as an isolated agent. The specific group assigned to an agent a is denoted as $G_a \in \{G^i\}$. We also compute the velocity of a group G as the average velocity of all agents belonging to G, and it is denoted as $\mathbf{v}_G$. During the simulation, the number of agents in a group G may change or may maintain the group formation. For example, nearby agents with similar goals and similar directions of motion will merge into a group and maintain that group by following one after another. A large group may split into several sub-groups while facing an obstacle or other groups, and these sub-groups may merge into one group after passing the obstacle. As two groups come close to each other, it is possible that agents may switch from one group to the other (i.e. reassign groups for an agent). As a result, it is important to support such group behaviors corresponding to formation, merging, splitting, reassignment, etc.

3.2. Agent-Group Velocity Obstacle

For collision avoidance between the agents, we use the concept of velocity obstacles [vdBGLM11]. In order to perform collision avoidance between groups, we use the notion of velocity obstacle $VO_{a|G}$ for one agent a induced by a group G. Given the velocity of the group $\mathbf{v}_G$, $VO_{a|G}$ can be defined as the set of agent a's velocities $\mathbf{v}_a$ that will result in a collision with G at some point within the time window τ, assuming that the group G maintains its velocity $\mathbf{v}_G$ during τ:

$$VO^{\tau}_{a|G} = \{\mathbf{v} | \exists t \in [0,\tau] \text{ such that } \mathbf{p}_a + (\mathbf{v} - \mathbf{v}_G)t \in \mathcal{CH}(G) \oplus \mathcal{D}(\mathbf{0}, r_a)\}, \quad (1)$$

where $\mathcal{D}(\mathbf{0}, r_a)$ is a disc centered at the origin with radius r_a, and $\mathcal{CH}(G)$ is convex hull of the set of agents constituting the group G. The convex hull provides a conservative bound that can guarantee collision-free navigation. This equation implies that if agent a chooses a velocity outside the velocity obstacle $VO_{a|G}$, it will not collide with group G within the time window τ. Intuitively, the velocity obstacle can be geometrically constructed as a cone with its apex in $\mathbf{p}_a$ and its sides tangent to $\mathcal{CH}(G)$ expanded by the radius r_a

of the agent a, which is then translated by $\mathbf{v}_G$, as shown in Figure 4. From the geometric interpretation, we can observe that the convex hull $\mathcal{CH}(G)$ need not be computed explicitly. Instead, the velocity obstacle can be fully defined by the extreme agents in the radial directions of the group, as observed from $\mathbf{p}_a$. We denote the most "clockwise" agent as e_a^r and the most "counterclockwise" agent as e_a^l. These two agents e_a^r and e_a^l are used to compute a collision free trajectory for agent a.

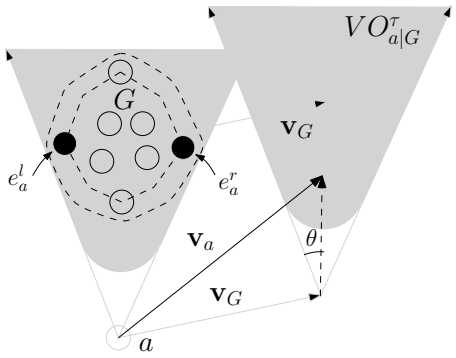


Figure 4: Agent-group Velocity Obstacle: The velocity obstacle $VO^{\tau}_{a|G}$ for agent a induced by a group G of agents. The group G contains seven agents and its convex hull is the dashed line. If G only contains a single agent, $VO_{a|G}$ reduces to the traditional velocity obstacle between two agents. The black agents e_a^l and e_a^r are the two most extreme agents in the group G. The angle θ represents the steering angle required by agent a to avoid the group of agents G.

3.3. Our Approach

Our goal is to generate realistic dynamic grouping behaviors for pedestrians. We assign the agents to a different group during each frame and compute the agents' trajectories by taking into account group dynamics. In cluttered areas, the agents tend to be assigned to a large group, and in open areas the agents tend to be well spread out and may not be assigned to any group. As a result, we need the capability to support such dynamic merging and splitting behaviors. Furthermore, our approach is motivated by the principle of least effort [GCC*10] that has been used for computing the agent trajectories in prior crowd simulation algorithms. An agent aligns itself with a group such that the resulting motion is governed by effort minimization. In particular, given the preferred velocity for each agent a, we tend to compute the actual velocity that tends to minimize the effort required by the entire group G_a to avoid the obstacles. In order to support dynamic groups, our approach supports the following computations:

Group formation: We use spatial clustering algorithms to generate the initial group assignment for each agent in the crowd. The isolated agents are not assigned to any group.

Group maintenance and navigation: Our approach tends to maintain these groups as long as possible during the navigation. All of the agents belonging to a group exhibit coherent trajectories and behaviors. We perform inter-group and intra-group computations to generate such behaviors. At the inter-group level, each group needs to perform high-level coherent trajectory computations to avoid collisions with other groups and obstacles. The collision avoidance policy is chosen in a manner such that if all agents in the same group consistently make the same choice, the entire group tends to avoid other groups altogether. At the intra-group level, each agent inside a group (except the group leader) will choose one fellow agent from the same group and follow it to make progress towards the goal. If each agent in the group follows this policy, our approach doesn't need to explicitly check for agent-agent collisions within a group.

Group update: The group assignments are updated and the number of agents belonging to a group may change.

A key component for trajectory computation is an efficient group-group collision avoidance algorithm. In our approach, this is performed by first avoiding the collisions between the group leader of each group and other groups, and then determining a suitable adapted velocity for other non-leader agents. In particular, we use the ORCA-based agent-group collision avoidance technique [vdBGLM11] to compute the current velocity for the group leader. All of the other agents in the same group will compute their velocity according to the following policy. The new adapted velocity for each agent is used by the agent-agent ORCA algorithm to compute the actual velocities for each agent by taking into account all the constraints. The preferred velocity is chosen such that it guides the agent towards its goal position.

4. Multi-Agent Simulation

In this section, we present our multi-agent simulation algorithm that can simulate dynamic grouping behaviors.

4.1. Group Formation

We use a spatial clustering algorithm to compute the initial group assignment for each agent. This assignment is based on the positions and velocities of all agents. The clustering criteria are based on the following criteria. A pair of agents, a and b, they belong to the same group if the following conditions hold:

- the position $\mathbf{p}_a$ of agent a and the position $\mathbf{p}_b$ of agent b are within a predefined distance ε_p, and
- the velocity $\mathbf{v}_a$ of agent a and the velocity $\mathbf{v}_b$ of agent b are within a predefined threshold ε_v.

The transitive closure of this relation uniquely classifies each cluster into groups, and can be formally described as

$$(a \sim b) \equiv (\|\mathbf{p}_b - \mathbf{p}_a\| < \varepsilon_p \wedge \|\mathbf{v}_b - \mathbf{v}_a\| < \varepsilon_v),$$

where $\sim$ is the binary operator defining whether two agents would be grouped together. Given this criterion for grouping, we use a greedy algorithm to compute these groups $\{G^i\}$ in $\mathcal{O}(n)$ time, where n are the number of agents in the crowd. In particular, we iteratively check each agent as to whether or not it can be grouped into any existing groups according to the $\sim$ relationship. If an agent is not assigned to any group, it is treated as a single or isolated agent during that frame.

4.2. Group Maintenance and Navigation

One key point in simulating the group behavior for a crowd is maintaining the groups based on collision avoidance constraints during the navigation. We achieve group maintenance by using a two-level approach: the inter-group level makes sure that the entire group will avoid other groups as a whole and with the least effort, and the intra-group level ensures that all the agents belonging to a group do not collide with each other.

4.2.1. Inter-Group Level Behaviors

In most multi-agent simulation algorithms, each agent independently computes its current velocity for collision avoidance. However, such navigation algorithms may not be able to maintain the group-like coherent motion. This is because each agent may choose different extreme agents (as shown in Figure 4) from the same group to avoid collision, due to their difference in positions and velocities relative to the obstacle group. Instead, we prefer that each agent in the same group as a should select the identical side (all e_l or e_r) while bypassing one group G. In addition, the bypassing side should also follow the least effort principle, i.e., the group of agents should bypass other agents with the minimum effort.

For this purpose, we first estimate the effort required for agent a to bypass one obstacle group G as

$$E_a = (\mathbf{v}_a - \mathbf{v}_G) \times (\mathbf{p}_a - \mathbf{p}_G) \cdot \mathbf{n}, \tag{2}$$

where $\mathbf{v}_G$ and $\mathbf{p}_G$ are the average velocity and position of the group G, respectively, and $\mathbf{n}$ is the normal of the 2D plane. As shown in Figure 4, this effort measurement is the sine function with the steering angle θ required by the agent to avoid collision with the obstacle group. Then the total effort for the entire group G_a can be computed as $E = \sum_{b \in G_a} E_b$, and the bypassing side (for navigation) is computed as:

$$s = \begin{cases} \text{r (right)} & \text{if } E < 0 \\ \text{l (left)} & \text{otherwise.} \end{cases} \tag{3}$$

In other words, each agent would choose the same bypassing side which requires a smaller effort for collision avoidance. Given the bypassing side, we also compute a's adapted velocity $\mathbf{v}_a^{\text{adapted}}$ such that the agent a can avoid this group G with the least effort. The algorithm is as shown in Algorithm 1. Note that the agents in the same group may choose different extreme agents. For instance, as shown in Figure 5, the agents c and e are in the same group, but their choice of extreme agents are h and g respectively.

4.2.2. Intra-Group Level Behaviors

Even when the agents in the same group select the same bypassing side, they may avoid the extreme agents in a different manner. This is due to the fact that the local navigation algorithm does not consider the group-level information. For instance, in Figure 5, agents c and d have the same extreme agent h. According to the local navigation, the agent c will avoid h from the left side while d will avoid h from the right side. This leads to a quick splitting of a group of agents. To solve this problem, we use the *dynamic following strategy*. In particular, as shown in Algorithm 2, each agent $a \in G$ will choose to follow an agent b^* which is computed as

$$b^* = \underset{\|\mathbf{p}_b - \mathbf{p}_a\|}{\arg\min} \{b | b \in G, F(a,b) > 0, \|\mathbf{p}_a - e_a^s\| > \|\mathbf{p}_b - e_a^s\|\}, \tag{4}$$

where $F(a,b) = [(\mathbf{p}_b - \mathbf{p}_a) \times (e_a^s - \mathbf{p}_a) \cdot \mathbf{n}] \cdot E$. This implies that by following the agent b, the agent a will bypass the extreme agent e_a^s from the correct bypassing side. The condition $\|\mathbf{p}_a - e_a^s\| > \|\mathbf{p}_b - e_a^s\|$ implies that e_a^s should be closer to the agent b, which is to be followed than the agent a. $\arg\min_{\|p_b - p_a\|}$ is used to specify that a intends to follow its closest neighbor. For the agents that cannot find a suitable agent to follow, tend to become the leaders of the group. For instance, in Figure 5, both a and f act as leaders while all other agents in the red group act as followers.

In addition, to encourage the agents to move forward rather than bypassing other agents, we require that the adapted velocity should be in the same direction as the preferred velocity, i.e. $\mathbf{v}_a^{\text{adapted}} \cdot \mathbf{v}_a^{\text{pref}} > 0$.

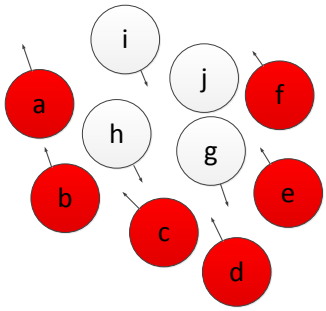


Figure 5: Leveraging the following-leading strategy to automatically generate splitting-merging behavior. In this scenario, the agents a and f are two leaders for the group of red agents, which will be split into two separate groups. The group of white agents will move forward coherently and pass through the red group.

4.3. Group Update

The group update or reassignment happens under two situations. The first situation occures while the agents are in an open area and can easily approach their goals. In this case, the group bypassing and dynamic following strategies are usually sub-optimal for an individual agent's trajectory, even though they are beneficial for the overall navigation. As a result, the notion of being able to stop following at a suitable time will help improve the performance of the multi-agent navigation system. We perform this step by checking whether or not the original preferred velocity $\mathbf{v}_{\text{pref}}$ will result in making the agent collide with any other agents. If not, the agent will detach from the group and will use the discrete agent local navigation algorithm based on ORCA to move towards its goal.

The second situation arises when the current group setting is not able to compute a collision-free velocity for the navigation. This is mainly because the original groups have deformed too much during the navigation, and their shapes have become quite non-convex. Our solution is to perform re-clustering over the entire crowd, to generate a group partition that can better describe the current dynamic behavior of the pedestrian crowd.

Algorithm 1: Inter-group level group maintenance – computing the adpated velocity for each agent

input : A group G and all other groups $\{G_i\}$;
output: The extreme agents e that all agents $a \in G$ needs to bypass, and a's adapted velocities $\mathbf{v}_a^{\text{adapted}}$

for each agent $a \in G$ **do**
  $\mathbf{v}_a^{\text{adapted}} \leftarrow nil$
  for each group $G' \in \{G_i\}$ **do**
    Compute G''s two extreme agents e_a^l and e_a^r to a
    Compute E according to Equation 2
Determine the bypassing side s of G according to Equation 3
`/* Using the extreme agents and the bypass side to calculate the adapted velocity for each agent in G */`
for each agent $a \in G$ **do**
  if $v_a^{\text{adapted}} = nil$ or agent a collides with G at velocity v_a^{adapted} **then**
    Update the adapted velocity $\mathbf{v}_a^{\text{adapted}}$ by avoiding the group velocity obstacle $VO^{\tau}_{a|G}$ according to Equation 1.

Algorithm 2: Intra-group level group maintenance – computing the following strategy for the agent a in group G.

input : Agent a's adapted velocity $\mathbf{v}_a^{\text{adapted}}$ and its associated group G
output: The partner that the agent a chooses to follow

for each agent b in the group G except a **do**
  $\tilde{\mathbf{v}}_a^{\text{adapted}} = \frac{\mathbf{p}_b - \mathbf{p}_a}{\|\mathbf{p}_b - \mathbf{p}_a\|} v_{max}$
  if b satisfies Equation 4 and $\mathbf{v}_a^{\text{adapted}} \cdot \mathbf{v}_a^{\text{pref}} > 0$ **then**
    $\mathbf{v}_a^{\text{adapted}} \leftarrow \tilde{\mathbf{v}}_a^{\text{adapted}}$
    a chooses b as the follower
  else
    a acts as the leader.

5. Implementation and Performance

In this section we describe our implementation and highlight the performance of our algorithm on different benchmarks. We compare our result with the grouping behaviors generated using state-of-the-art crowd simulators: the agent-agent collision avoidance algorithm ORCA [vdBGLM11], a group-based *meso-scale* navigation approach [HvdB13], and recent work on implicit navigation [GKGG16]. We use five benchmarks to evaluate our algorithms. Three of them are designed from real-world videos, and we compare the movement trajectories generated by different approaches. Two others are synthetic benchmarks in which we also compare the running time and the number of collisions between the agents during the simulation. We have implemented our algorithms in C++ on an Intel Core i7 CPU running at 3.30GHz with 16GB of RAM and running Windows 7. All of the timing results are generated on a single core.

5.1. Real-World Scenarios and Validation

We compare the crowd simulation results using our dynamic group behavior generation algorithm and prior approaches on scenarios inspired by real-world crowd videos. We extract the trajectories of the agents in the real-world videos using a pedestrian tracking algorithm [BKM15]. For each crowd simulation algorithm, the number of agents and their initial positions and goal positions are assigned according to the pedestrian tracking results. Given the initial and goal positions, we compare the trajectories of the pedestrians generated by each algorithm and compare them with those in the real videos in Figure 10. Figures 6 and 7 show the key frame for simulation sequences generated using different approaches. We can observe that the simulation results using our dynamic group generation algorithm are most similar to the real world pedestrians in terms of trajectory behaviors.

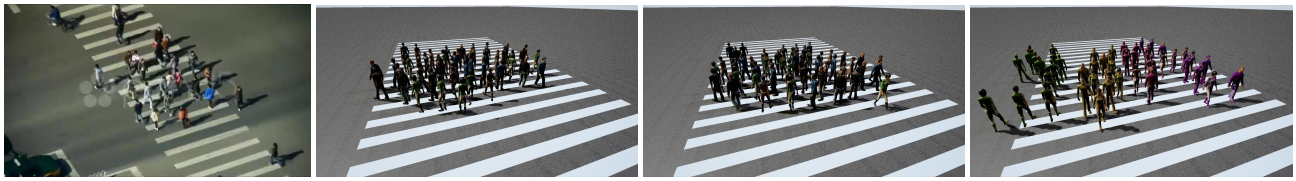
In terms of quantitative comparison, we evaluate the behavior of real pedestrians with that of simulated crowds by comparing:

1. Compare the running time and number of collisions that occurred during the navigation from the initial to the goal positions, as shown in Table 1, and
2. Compare the trajectories extracted using the tracking algorithm (i.e. the ground truth) for some of the agents with the trajectories computed by different multi-agent simulation algorithms.

In the first benchmark, agents are passing through a crosswalk as shown in Figure 1. During this simulation, agents automatically aggregate into groups and perform group-level collision avoidance. In this benchmark, the total time taken by different crowd simulation approaches is almost similar. However, our dynamic group behavior approach results in fewer collisions between the agents during navigation. Moreover, the trajectories generated using our algorithm have a better match with the ground truth data, as shown in Figure 6. This is due to the fact that ORCA and meso-scale simulation algorithms need more space to perform collision avoidance and therefore the agents are more spread out.

In the second benchmark, agents are moving in a hallway inside the building, which represents a tight space. In this simulation, each agent's initial position and direction of movement is computed based on the real-world trajectories. Our approach can compute the navigation trajectories with a fewer collisions and with coherent grouping behaviors, similar to real-world videos. In contrast, the agents in ORCA and meso-scale simulation algorithms take more time to move from the initial to the goal positions due to the tight spaces. Moreover, the trajectories computed by our algorithm are smoother and there is a high co-relation with the ground truth data, i.e. the extracted trajectories.

The third benchmark corresponds to a cluttered environment where the agents need to go through the hallway, as shown in Figure 9. Both RVO and meso-scale methods that are unable to compute collision-free navigation as the crowd density is high. Instead, our method automatically enables the agents to move in groups and compute collision-free trajectories. We also observe that the trajec-



(a) real-world video frame (b) ORCA (c) meso-scale (d) our method

Figure 6: For benchmark 1, we compare the group behavior generated by our algorithm (d) on a real-world scenario (a). As compared to ORCA (b) and meso-scale (c), our approach can generate smoother and more coherent trajectories.

(a) real-world video frame (b) ORCA (c) meso-scale (d) our method

Figure 7: Comparison between the key frame for simulation sequences generated using different approaches on benchmark 3. Our method exhibits group behaviors as seen in the captured real-world video.

tories computed using our algorithm have a better match with the ground truth data.

5.2. Other Benchmarks

We also generated some synthetic scenes to further evaluate the performance of our dynamic group behavior generation algorithm. In the fourth benchmark, agents are randomly placed in the scenario. Our approach automatically clusters them into groups and generates coherent trajectories. Furthermore, it results in fewer collisions and smoother trajectories. The fifth benchmark corresponds to adding several static obstacles in the environment corresponding to the fourth benchmark. Our method can compute the paths to the goal position for each agent. On the other hand, the agents get stuck and pushed away from the goal position within ORCA and meso-scale simulation

6. Limitations, Conclusions and Future Work

We present a novel multi-agent navigation algorithm that can automatically generate dynamical grouping behaviors. Our approach is general and makes no assumptions about the size or shape of the group, and can dynamically adapt to the environment. Moreover, it results in smooth and coherent navigation behaviors as compared to prior multi-agent reciprocal collision avoidance algorithms. Furthermore, the agent's tend to avoid congestion based on the group's follow-the-leader trajectory computation behavior, which is similar to human behaviors observed in real-world scenarios. We demonstrate its performance on complex benchmarks with a few hundred agents and show that the trajectories generated by our algorithm are similar to those observed in real-world scenarios and exhibit similar group behaviors. Unlike prior group behavior simulation schemes, our approach is adaptive and can model the dynamic behaviors of the agent in response to the environment.

Our approach has some limitations. It is currently designed for homogeneous agents and the clustering algorithm only takes into account the position and velocity of each agent. We don't account for agents with varying personalities or how they respond to the environmental effects, situations, the psychological component corresponding to the concept of personal space that varies among different cultures, or the social norms. Our reciprocal group-group collision avoidance algorithm can be conservative as it is implicitly based on the convex hull or extreme agents.

There are many avenues for future work. In addition to overcoming these limitations, we would like to evaluate its performance in complex scenarios with tens of thousands of agents (e.g. sporting events or religious gatherings). We would like to further validate its performance using other metrics, such as comparing it with the collective behaviors and fundamental diagrams of real-world crowds. Finally, we would like to combine with macroscopic techniques to simulate very dense crowds.

7. Acknowledgements

This research is supported in part by ARO Contract W911NF-10-1-0506, NSF award 1305286, RGC GRF 17204115, and a grant from Boeing.

References

[AMBR*12] ALONSO-MORA J., BREITENMOSER A., RUFLI M., SIEGWART R., BEARDSLEY P.: Image and animation display with multiple mobile robots. *International Journal of Robotics Research 31*, 6 (2012), 753–773. 2

[BKHF14] BERSETH G., KAPADIA M., HAWORTH B., FALOUTSOS P.: Steerfit: Automated parameter fitting for steering algorithms. In *Symposium on Computer Animation* (2014). 2

[BKM15] BERA A., KIM S., MANOCHA D.: Efficient trajectory extraction and parameter learning for data-driven crowd simulation. In *Graphics Interface* (2015), pp. 65–72. 6

[CJ61] COLEMAN J. S., JAMES J.: The equilibrium size distribution of freely-forming groups. *Sociometry* (1961), 36–45. 1

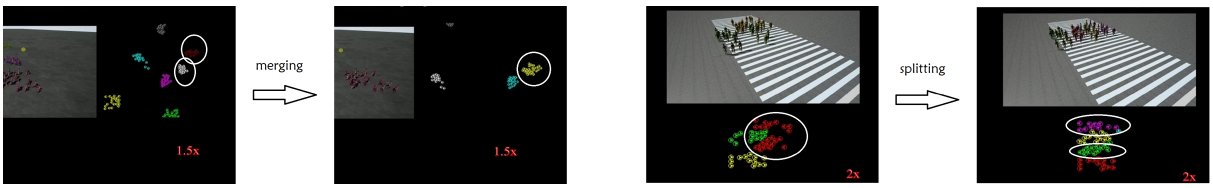


Figure 8: Key frames demonstrating the merge and split behaviors generated by our approach. In the first two figures (left), two groups merge together; In the second two figures (right), a group of agents are split into two groups.

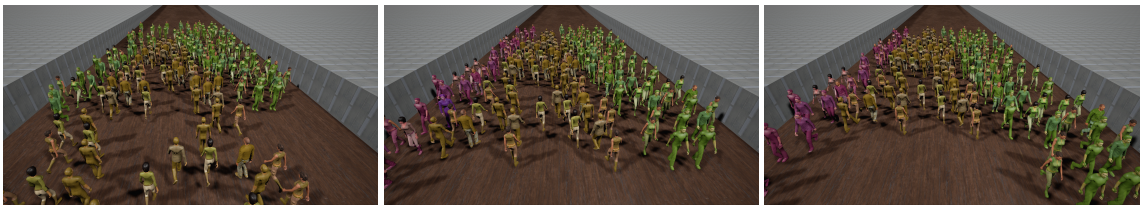


Figure 9: Key frames for the simulation sequence generated by our algorithm on benchmark 2. Our approach results in smoother trajectories and fewer inter-agent collisions.

[CSM12] CURTIS S., SNAPE J., MANOCHA D.: Way portals: Efficient multi-agent navigation with line-segment goals. In *Symposium on Interactive 3D Graphics and Games* (2012), pp. 15–22. 2

[DGABar] DURUPINAR F., GUDUKBAY U., AMAN A., BADLER N. I.: Psychological parameters for crowd simulation: from audiences to mobs. *IEEE Transactions on Visualization and Computer Graphics* (2015, to appear). 2

[GBS14] GORRINI A., BANDINI S., SARVI M.: Group dynamics in pedestrian crowds: Estimating proxemic behavior. *Transportation Research Record: Journal of the Transportation Research Board*, 2421 (2014), 51–56. 1, 2

[GCC*10] GUY S. J., CHHUGANI J., CURTIS S., DUBEY P., LIN M., MANOCHA D.: Pledestrians: a least-effort approach to crowd simulation. In *Symposium on computer animation* (2010), pp. 119–128. 4

[GCK*09] GUY S. J., CHHUGANI J., KIM C., SATISH N., LIN M., MANOCHA D., DUBEY P.: Clearpath: highly parallel collision avoidance for multi-agent simulation. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), ACM, pp. 177–187. 2

[GKGG16] GODOY J., KARAMOUZAS I., GUY S. J., GINI M.: Implicit coordination in crowded multi-agent navigation. *AAAI Conference on Artificial Intelligence* (2016). 2, 6

[GNCL14] GOLAS A., NARAIN R., CURTIS S., LIN M. C.: Hybrid long-range collision avoidance for crowd simulation. *IEEE Transactions on Visualization and Computer Graphics 20*, 7 (2014), 1022–1034. 2

[HKBK14] HUANG T., KAPADIA M., BADLER N. I., KALLMANN M.: Path planning for coherent and persistent groups. In *IEEE International Conference on Robotics and Automation* (2014), pp. 1652–1659. 2

[HLLO10] HUERRE S., LEE J., LIN M., O'SULLIVAN C.: Simulating believable crowd and group behaviors. In *ACM SIGGRAPH ASIA 2010 Courses* (2010). 2

[HM95] HELBING D., MOLNAR P.: Social force model for pedestrian dynamics. *Physical review E* (1995). 2

[HPWMar] HE L., PAN J., WANG W., MANOCHA D.: Proxemic group behaviors using reciprocal multi-agent navigation. In *IEEE International Conference on Robotics and Automation* (2015, to appear). 2

[HvdB13] HE L., VAN DEN BERG J.: Meso-scale planning for multi-agent navigation. In *IEEE International Conference on Robotics and Automation* (2013), pp. 2839–2844. 2, 6

[Jam53] JAMES J.: The distribution of free-forming small group size. *American Sociological Review* (1953). 1

[KCT*13] KAPADIA M., CHIANG I.-K., THOMAS T., BADLER N. I., KIDER JR. J. T.: Efficient motion retrieval in large motion databases. In *Symposium on Interactive 3D Graphics and Games* (2013), pp. 19–28. 2

[KDB12] KIMMEL A., DOBSON A., BEKRIS K.: Maintaining team coherence under the velocity obstacle framework. In *International Conference on Autonomous Agents and Multiagent Systems-Volume 1* (2012), pp. 247–256. 2

[KG15] KARAMOUZAS I., GUY S.: Prioritized group navigation with formation velocity obstacles. In *IEEE International Conference on Robotics and Automation* (2015), pp. 5983–5989. 2

[KLB12] KRONTIRIS A., LOUIS S., BEKRIS K.: Multi-level formation roadmaps for collision-free dynamic shape changes with non-holonomic teams. In *IEEE International Conference on Robotics and Automation* (2012), pp. 1570–1575. 2

[Kno73] KNOWLES E. S.: Boundaries around group interaction: The effect of group size and member status on boundary permeability. *Journal of Personality and Social Psychology 26*, 3 (1973), 327. 2

[KO04] KAMPHUIS A., OVERMARS M. H.: Finding paths for coherent groups using clearance. In *SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 19–28. 2

[KO12] KARAMOUZAS I., OVERMARS M.: Simulating and evaluating the local behavior of small pedestrian groups. *IEEE Transactions on Visualization and Computer Graphics 18*, 3 (2012), 394–406. 2

[LCHL07] LEE K. H., CHOI M. G., HONG Q., LEE J.: Group behavior from video: a data-driven approach to crowd simulation. In *Symposium on Computer Animation* (2007), pp. 109–118. 2

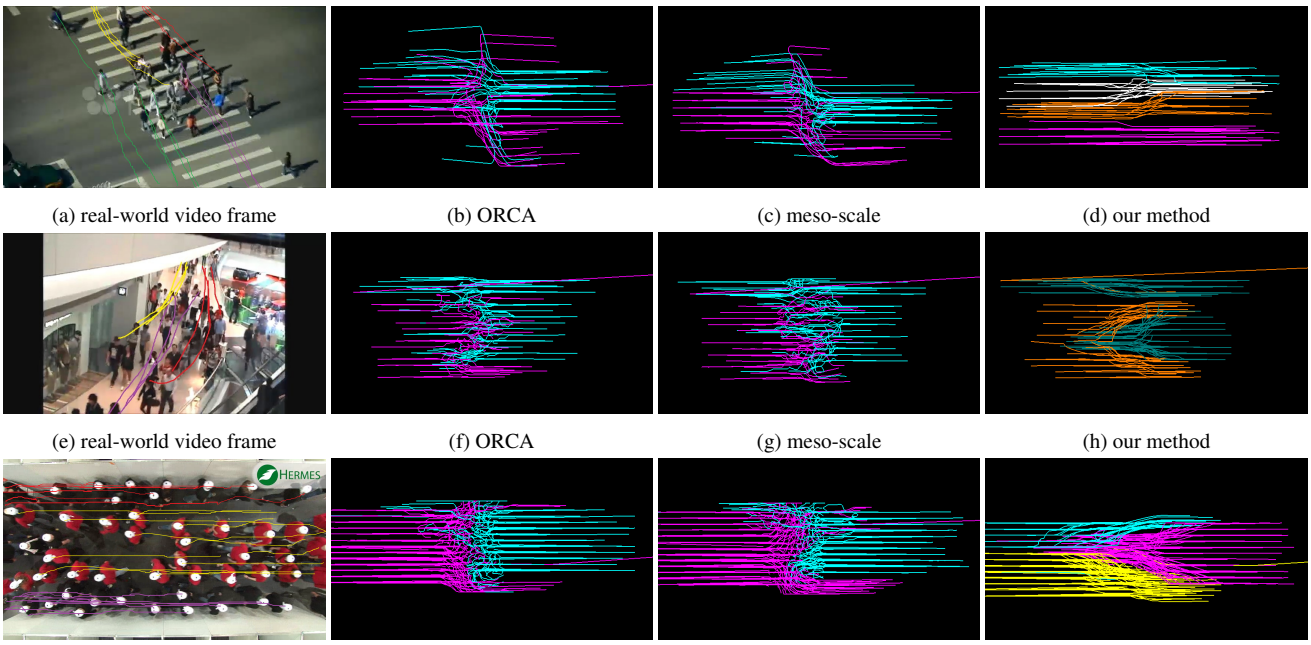
[LCSCO09] LERNER A., CHRYSANTHOU Y., SHAMIR A., COHEN-OR D.: Data driven evaluation of crowds. In *Motion in Games* (2009), pp. 75–83. 2

[NBCM15] NARANG S., BEST A., CURTIS S., MANOCHA D.: Generating pedestrian trajectories consistent with the fundamental diagram based on physiological and psychological factors. *PLoS one 10*, 4 (2015), e0117856. 2

[NGCL09] NARAIN R., GOLAS A., CURTIS S., LIN M. C.: Aggregate dynamics for dense crowd simulation. *ACM Transactions on Graphics 28*, 5 (2009), 122:1–122:8. 2

[OPOD10] ONDŘEJ J., PETTRÉ J., OLIVIER A.-H., DONIKIAN S.: A synthetic-vision based steering approach for crowd simulation. *ACM Transactions on Graphics 29*, 4 (2010), 123:1–123:9. 2

[PAB07] PELECHANO N., ALLBECK J. M., BADLER N. I.: Controlling



(a) real-world video frame (b) ORCA (c) meso-scale (d) our method

(e) real-world video frame (f) ORCA (g) meso-scale (h) our method

(i) real-world video frame (j) ORCA (k) meso-scale (l) our method

Figure 10: For three real-world benchmarks (the 1st column), we compare the trajectory behaviors generated by our algorithm (4th column, each color represents a group). As compared to ORCA (2nd column) and meso-scale (3rd column), our approach can generate smoother and coherent trajectories.

Method	Benchmark 1			Benchmark 2			Benchmark 3			Benchmark 4			Benchmark 5		
	avg frm	#steps	#colls	avg frm	#steps	#colls	avg frm	#steps	#colls	avg frm	#steps	#colls	avg frm	#steps	#colls
ORCA [vdBGLM11]	311	162	78	317	363	103	383	5000+	500+	319	209	257	321	5000+	500+
Meso-scale [HvdB13]	187	171	53	184	475	111	152	5000+	500+	97	212	262	93	5000+	500+
Proxemic [HPWM15]	151	169	1	121	257	2	103	691	3	107	237	1	91	403	1
Our (dynamic grouping)	263	161	2	287	221	6	257	253	3	201	203	3	183	387	2

Table 1: Performance Comparisons: We compare our approach with previous methods (ORCA, meso-scale, Proxemic) on five benchmarks. We report the performance in terms of frame rate (frm), the average number of simulation time steps taken for each agent to reach the goal position (#steps) and the average number of pairwise collisions between the aagents (#colls). These collisions can occur when the conservative collision avoidance schemes can't compute a feasible solution. In some case, the agents in the ORCA or meso-scale algorithms get stuck resulting in a high number of collisions. Even after 5000 simulation they have not reached the goal positions. We observe these behaviors with ORCA and meso-scale algorithms on Benchmark 3 and Benchmark 5. Proxemic only performs local clustering and can't generate splitting and reformation behaviors in large groups.

individual agents in high-density crowd simulation. In *Symposium on Computer animation* (2007), pp. 99–108. 2

[PQC12] PARK S. I., QUEK F., CAO Y.: Modeling social groups in crowds using common ground theory. In *Winter Simulation Conference* (2012), Winter Simulation Conference, p. 113. 2

[Rei01] REICHER S.: The psychology of crowd dynamics. *Blackwell handbook of social psychology: Group processes* (2001), 182–208. 3

[Rey87] REYNOLDS C. W.: Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH* (1987), pp. 25–34. 2

[SPN08] SILVEIRA R., PRESTES E., NEDEL L. P.: Managing coherent groups. *Computer Animation and Virtual Worlds 19*, 3-4 (2008), 295–305. 2

[SSKF10] SCHUERMAN M., SINGH S., KAPADIA M., FALOUTSOS P.: Situation agents: agent-based externalized steering logic. *Computer Animation and Virtual Worlds 21*, 3-4 (2010), 267–276. 2

[TCP06] TREUILLE A., COOPER S., POPOVIĆ Z.: Continuum crowds. In *SIGGRAPH* (2006), pp. 1160–1168. 2

[TYK*09] TAKAHASHI S., YOSHIDA K., KWON T., LEE K. H., LEE J., SHIN S. Y.: Spectral-based group formation control. *Computer Graphics Forum 28*, 2 (2009), 639–648. 2

[vdBGLM11] VAN DEN BERG J., GUY S., LIN M., MANOCHA D.: Reciprocal n-body collision avoidance. In *Robotics Research*, Pradalier C., Siegwart R., Hirzinger G., (Eds.), vol. 70 of *Springer Tracts in Advanced Robotics*. Springer Berlin Heidelberg, 2011, pp. 3–19. 2, 3, 4, 6

[WGO*14] WOLINSKI D., GUY S. J., OLIVIER A.-H., LIN M. C., MANOCHA D., PETTRÉ J.: Parameter estimation and comparative evaluation of crowd simulations. In *Eurographics* (2014). 2

[YCP*08] YEH H., CURTIS S., PATIL S., VAN DEN BERG J., MANOCHA D., LIN M.: Composite agents. In *Symposium on Computer Animation* (2008), pp. 39–47. 2

[YT07] YU Q., TERZOPOULOS D.: A decision network framework for the behavioral animation of virtual humans. In *Symposium on Computer animation* (2007), pp. 119–128. 2